Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 0, no. 0, pp. 0–0 (0)
DOI: 10.7155/jgaa.00534

Bundled Crossings Revisited

Steven Chaplick[1,2] *Thomas C. van Dijk*[1] *Myroslav Kryven*[1]
Ji-won Park[4] *Alexander Ravsky*[3] *Alexander Wolff*[1]

[1]Universität Würzburg, Würzburg, Germany
[2]Maastricht University, Maastricht, the Netherlands
[3]Pidstryhach Institute for Applied Problems of Mechanics and Mathematics, National Academy of Sciences of Ukraine, Lviv, Ukraine
[4]Université de Lorraine, CNRS, Inria, LORIA, Nancy, France

Abstract

An effective way to reduce clutter in a graph drawing that has (many) crossings is to group edges that travel in parallel into *bundles*. Each edge can participate in many such bundles. Any crossing in this bundled graph occurs between two bundles, i.e., as a *bundled crossing*. We consider the problem of bundled crossing minimization: A graph is given and the goal is to find a bundled drawing with at most k bundled crossings. We show that the problem is NP-hard when we require a simple drawing. Our main result is an FPT algorithm (in k) for simple *circular* layouts where vertices must be placed on a circle and edges must be drawn inside the circle. These results make use of the connection between bundled crossings and graph genus. We also consider bundling crossings in a given drawing, in particular for storyline visualizations.

Submitted:	Reviewed:	Revised:	Accepted:
November 2019	January 2020	April 2020	June 2020
	Final: June 2020	Published:	
Article type: regular paper		Communicated by: D. Archambault and C. Tóth	

A preliminary version of this paper appeared in Proc. 27th Int. Symp. Graph Drawing & Network Vis. [9]. S.C. acknowledges support by the joint DFG–GAČR grant WO 758/11-1; M.K. and A.R. acknowledge support by the joint DFG–SFFRU grant WO 758/9-1.

E-mail addresses: myroslav.kryven@uni-wuerzburg.de (Myroslav Kryven) alexander.ravsky@uni-wuerzburg.de (Alexander Ravsky)

1 Introduction

In traditional node–link diagrams, vertices are mapped to points in the plane and edges are usually drawn as straight-line segments connecting the vertices. For large and dense graphs, however, such layouts tend to be so cluttered that it is hard to see any structure in the data. For this reason, Holten [19] introduced *bundled drawings*, where edges that are close together and roughly go into the same direction are drawn using Bézier curves such that the grouping becomes visible. Due to the practical effectiveness of this approach, it has quickly been adopted by the InfoVis community [12,18,20,21,30]. However, bundled drawings have only recently attracted study from a theoretical point of view [2,16,17,35].

Crossing minimization is a fundamental problem in graph drawing [32]. Its natural generalization in bundled drawings is bundled crossing minimization, see Definition 1 for the formalization of a bundled crossing. In his survey on crossing minimization, Schaefer lists the bundled crossing number as a variant of the crossing number and suggests to study it [32, page 35].

Related Work. Fink et al. [17] considered bundled crossings (which they called block crossings) in the context of drawing metro maps. A metro network is a planar graph where vertices are stations and metro lines are simple paths in this graph. These paths representing metro lines can share edges. They enter an edge at one endpoint in some linear order, follow the edge as x-monotone curves (considering the edge as horizontal), and then leave the edge at the other endpoint in some linear order. In order to improve the readability of metro maps, the authors suggested to bundle crossings. The authors then studied the problem of minimizing bundled crossings in such metro maps. Fink et al. also introduced *monotone* bundled crossing minimization where each pair of lines can intersect at most once. Later, Fink et al. [35] applied the concept of bundled crossings to drawing storyline visualizations. A storyline visualization is a set of x-monotone curves where the x-axis represents time in a story. Given a set of *meetings* (subsets of the curves that must be consecutive at given points in time), the task is to find a drawing that realizes the meetings and minimizes the number of bundled crossings. Fink et al. showed that, in this setting, minimizing bundled crossings is fixed-parameter tractable (FPT) in the number of curves and can be approximated in a restricted case. Van Dijk et al. [36] gave ILP and SAT formulations of the problem and evaluated these experimentally.

Our research builds on recent works of Fink et al. [16] and Alam et al. [2], who extended the notion of bundled crossings from sets of x-monotone curves to general drawings of graphs. We discuss their results in more detail soon.

The degenerate crossing number is defined by allowing more than two edges to intersect at the same point; several variants (one of which is also called the genus crossing number) have been studied [1,28,29,33]. The degenerate crossing number and the bundled crossing number might look completely different, but it turns out that the degenerate crossing number is closely related to the non-orientable genus [28] while the bundled crossing number is closely related to the orientable genus as we will see.

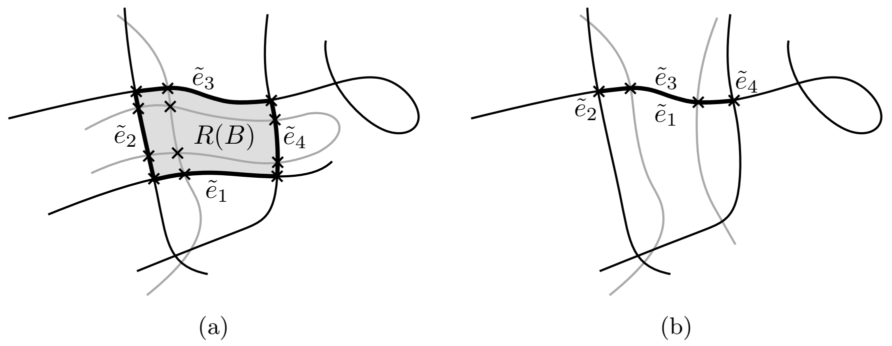


Figure 1: (a) A non-degenerate bundled crossing B and (b) a degenerate bundled crossing B'; crossings belonging to a bundled crossing are marked with crosses

Notation and Definitions. In graph drawing, it is common to define a drawing of a graph as a function that maps vertices to distinct points in the plane and edges to Jordan arcs that connect the corresponding points. In this paper, we are less restrictive; we sometimes allow edges to self-intersect. However, we forbid any three edges to share the same point. We will often identify vertices with their points and edges with their curves. Moreover, we assume that each pair of edges shares at most a finite number of points, that edges can touch (that is, be tangent to) each other only at endpoints, and that any point of the plane that is not a vertex is contained in at most two edges. A drawing is *simple* if any two edges intersect at most once and no edge self-intersects. We consider both simple and non-simple drawings; look ahead at Fig. 2 for a simple and a non-simple drawing of $K_{3,3}$.

Definition 1 (Bundled Crossing) *Let D be a drawing, not necessarily simple, and let $I(D)$ be the set of intersection points among the edges (not including the vertices) in D. We say that a* bundling *of D is a partition of $I(D)$ into* bundled crossings*, where a set $B \subseteq I(D)$ is a bundled crossing if the following holds (see Fig. 1).*

- *B is contained in a closed region $R(B)$ of the plane whose boundary consists of four Jordan arcs $\tilde{e}_1$, $\tilde{e}_2$, $\tilde{e}_3$, and $\tilde{e}_4$ that are pieces of edges e_1, e_2, e_3, and e_4 in D (with $\tilde{e}_i = e_i \cap R(B)$ for $i \in \{1,2,3,4\}$).*
- *The pieces of the edges cut out by the region $R(B)$ can be partitioned into two sets $\tilde{E}_1$ and $\tilde{E}_2$ such that $\tilde{e}_1, \tilde{e}_3 \in \tilde{E}_1$, $\tilde{e}_2, \tilde{e}_4 \in \tilde{E}_2$, and each pair of edge pieces in $\tilde{E}_1 \times \tilde{E}_2$ has exactly one intersection point in $R(B)$, whereas no two edge pieces in $\tilde{E}_1$ intersect and no two edge pieces in $\tilde{E}_2$ intersect.*

Our definition is similar to that of Alam et al. [2] but defines the Jordan region $R(B)$ more precisely. We call the sets $\tilde{E}_1$ and $\tilde{E}_2$ of edge pieces *bundles* and the Jordan arcs $\tilde{e}_1, \tilde{e}_3 \in \tilde{E}_1$ and $\tilde{e}_2, \tilde{e}_4 \in \tilde{E}_2$ *frame arcs* of the bundles $\tilde{E}_1$ and $\tilde{E}_2$, respectively. For simple drawings, we accordingly call the edges that bound the two bundles of a bundled crossing *frame edges*. We say that a bundled crossing is *degenerate* if at least one of the bundles consists of only one edge piece; see Fig. 1b. In this case, the region of the plane associated with the crossing coincides with that edge piece. In particular, any point in $I(D)$ by itself is a degenerate bundled crossing. Hence, every drawing admits a trivial bundling.

We use $\mathrm{bc}(G)$ to denote the *bundled crossing number* of a graph G, i.e., the smallest number of bundled crossings over all bundlings of all simple drawings of G. When we do not insist on simple drawings, we denote the corresponding number by $\mathrm{bc}'(G)$. In the circular setting, where vertices are required to lie on the boundary of a disk and edges inside this disk, we consider the analogous *circular bundled crossing numbers* $\mathrm{bc}^\circ(G)$ and $\mathrm{bc}^{\circ\prime}(G)$ of a graph G. If, in addition, the vertices are required to be in a prescribed circular order π, we consider the circular bundled crossing number with a *fixed vertex order* π and denote this number as $\mathrm{bc}^\circ(G,\pi)$. (In the literature on book embeddings, circular layouts are commonly called *1-page (book) layouts*, but since we do not consider more than one page here, we follow previous literature [2, 16] on the bundled crossing number and stick to the term *circular*.)

By $\mathrm{bc}(G,D)$ we denote the bundled crossing number of a specific simple drawing D of G. We say *fixed* drawing for this case. Similarly, by $\mathrm{bc}'(G,D)$ we denote the bundled crossing number of a specific, not necessarily simple drawing D of a graph G. By $\mathrm{bc}^\circ(G,D)$ we denote the bundled crossing number of a simple circular drawing D of a graph G. As Fink et al. [16] observed[1] in this variant of the problem, we can assume the graph to be a matching.

Fink et al. [16] showed that it is NP-hard to compute the minimum number $\mathrm{bc}(G,D)$ of bundled crossings that a given drawing D of a graph G can be partitioned into. They also showed that this problem generalizes the problem of partitioning a rectilinear polygon with holes into the minimum number of rectangles, and they exploited this connection to construct a 10-approximation for computing the number $\mathrm{bc}^\circ(G,D)$ of bundled crossings in the case of a circular drawing. They left open the computational complexity of the general and the circular bundled crossing number for the case that the drawing is not fixed.

Alam et al. [2] showed that $\mathrm{bc}'(G)$ equals the orientable genus of G, which in general is NP-hard to compute [34]. They also showed that there is a graph G with $\mathrm{bc}'(G) \neq \mathrm{bc}(G)$ by proving that $\mathrm{bc}'(K_6) = 1 < \mathrm{bc}(K_6)$. As it turns out, the two problem variants differ in the circular setting, too (see Fig. 2 and Observation 2). For computing $\mathrm{bc}(G)$ and $\mathrm{bc}^\circ(G)$, Alam et al. [2] gave an algorithm whose approximation factor depends on the density of the graph. They posed the existence of an FPT algorithm for $\mathrm{bc}^\circ(G)$ as an open question.

Our Contribution. As some graphs G have $\mathrm{bc}'(G) \neq \mathrm{bc}(G)$ (see Fig. 2), Fink et al. [16] posed the complexity of computing the bundled crossing number $\mathrm{bc}(G)$ of a given graph G as an open problem. We settle this in Section 2 as

[1] Fink et al. [16] used "embedding" and $\mathrm{bc}(\mathcal{E})$ where we use "drawing" and $\mathrm{bc}(G,D)$, respectively.

follows:

Theorem 1 *Given a graph G, it is NP-hard to compute* $\mathrm{bc}(G)$.

Our main result, which we prove in Section 3, resolves an open question of Alam et al. [2] concerning the fixed-parameter tractability of bundled crossing minimization in circular layouts as follows:

Theorem 2 *There is a computable function f such that, for any n-vertex graph G and integer k, we can check, in $O(f(k)n)$ time, whether* $\mathrm{bc}^\circ(G) \leq k$, *i.e., whether G admits a circular layout with k bundled crossings. Within the same time bound, such a layout can be computed.*

To prove this, we use an approach similar to that of Bannister and Eppstein [5] for 1-page crossing minimization (that is, edge crossing minimization in a circular layout). Bannister and Eppstein observe that the set of crossing edges of a circular layout with k edge crossings of a graph G forms an arrangement of curves that partition the drawing into $O(k)$ subgraphs, each of which occurs in a distinct face of this arrangement. The subgraphs are obviously outerplanar. This means that G has bounded treewidth. So, by enumerating all ways to draw the crossing edges of a circular layout with k edge crossings, and, for each such way, expressing the edge partition problem (into crossing edges and outerplanar components) in extended monadic second order logic (MSO_2), *Courcelle's Theorem* [10] (stated as Theorem 5 in Section 3) can be applied (leading to fixed-parameter tractability).

The difficulty in using this approach for bundled crossing minimization is in showing how to partition the graph into a set of $O(k)$ "crossing edges" (our analogy will be the frame edges) and a collection of $O(k)$ outerplanar graphs. This is where we exploit the connection to genus. Moreover, constructing an MSO_2 formula is somewhat more difficult in our case due to the more complex way our regions interact with our special set of edges.

Again using the above-mentioned connection, here between genus and the circular bundled crossing number $\mathrm{bc}^{\circ\prime}$, we can decide whether $\mathrm{bc}^{\circ\prime}(G) = k$ in $2^{O(k)}n$ time. In other words, if non-simple drawings are allowed, the problem is also FPT in k; see Section 3 (Theorem 4).

We also consider the setting where we are given a drawing and the task is to bundle the existing edge crossings into as few bundled crossings as possible, that is, computing $\mathrm{bc}(G, D)$ for a given drawing D of a graph G. We show in Section 4 that we can use an algorithm of Marx and Philipczuk [25, Theorem 1.3] (see page 21) to test whether $\mathrm{bc}(G, D) \leq k$ in $m^{O(\sqrt{k})}$ time for any simple drawing D with m edges. This yields an FPT-algorithm for testing whether $\mathrm{bc}^\circ(G, D) \leq k$ in $2^{O(\sqrt{k}\log k)} + O(m)$ time and for testing whether $\mathrm{bc}^\circ(G, \pi) \leq k$ in $2^{O(k^2)} + O(m)$ time, improving on an $\left(2^{O(k^2 \log k)} + O(m)\right)$-time algorithm of Alam et al. [2]

In Section 5 we consider storyline visualizations. In contrast to the above results, the storyline literature considers the number of characters m to be small and the number of crossings to be large. (Recall that storyline visualizations

Table 1: Algorithmic and complexity results concerning bundled crossing minimization for an m-edge graph G with restrictions such as vertex order π, drawing D, edge density δ, and k bundled crossings. We omit polynomial terms, and φ is the golden ratio. Our results are in boldface.

General layout			Circular layout		
$\mathrm{bc}(G)$	$\frac{6\delta}{\delta-3}$-approx. for $\delta>3$	[2]	$\mathrm{bc}^{\circ}(G)$	$\frac{6\delta}{\delta-2}$-approx. for $\delta>2$	[2]
	NP-hard	(Thm. 1)		**FPT**	(Thm. 2)
$\mathrm{bc}'(G)$	NP-hard	[16]	$\mathrm{bc}^{\circ\prime}(G)$	**FPT**: $2^{O(k)}$	(Thm. 4)
$\mathrm{bc}(G,D)$	NP-hard	[16]	$\mathrm{bc}^{\circ}(G,D)$	10-approximation	[16]
	XP: $m^{O(\sqrt{k})}$	(Thm. 6(b))		**FPT**: $k^{O(\sqrt{k})}$	(Thm. 6(c))
$\mathrm{bc}'(G,D)$	**XP**: $c^{O(\sqrt{k})}$	(Thm. 6(a))			
Storyline layout			$\mathrm{bc}^{\circ}(G,\pi)$	16-approx., FPT: $k^{O(k^2)}$	[2]
$\mathrm{bc}^{\mathrm{s}}(\mathrm{D})$	**FPT**: φ^{2m}	(Thm. 8)		**FPT**: $2^{O(k^2)}$	(Thm. 6(d))

are non-simple.) We show that computing the bundled crossing number $\mathrm{bc}^{\mathrm{s}}(D)$ of a given storyline visualization D can be done in $O(\varphi^{2m}\mathrm{poly}(m+c))$ time, where c is the number of crossings in D and φ is the golden ratio. Note that this is fixed parameter tractable in m.

For an overview of existing and new results see Table 1.

2 Computing bc(G) Is NP-Hard

For a given graph G, finding a drawing with the fewest bundled crossings resembles computing the *orientable genus*[2] $\mathrm{g}(G)$ of G. In fact, Alam et al. [2] showed that $\mathrm{bc}'(G) = \mathrm{g}(G)$. Thus, deciding whether $\mathrm{bc}'(G) = k$ for some k is NP-hard and FPT in k since the same holds for deciding whether $\mathrm{g}(G) = k$ [22, 27, 34].

Theorem 3 ([2]) *For every graph G with genus k, it holds that* $\mathrm{bc}'(G) = k$.

To show this, Alam et al. [2] first showed that a drawing with k bundled crossings can be lifted onto a surface of genus k, and thus $\mathrm{bc}'(G) \geq \mathrm{g}(G)$:

Observation 1 ([2]) *A drawing D with k bundled crossings can be* lifted *onto a surface of genus k via a one-to-one correspondence between bundled crossings and handles, i.e., at each bundled crossing, we attach a handle for one of the two edge bundles, thus providing a crossing-free* lifted drawing*; see Fig. 8.*

Then, to see that $\mathrm{bc}'(G) \leq \mathrm{g}(G)$, Alam et al. [2] used the *fundamental polygon* representation (or *polygonal schema*) [13] of a drawing on a genus-g surface. More precisely, the sides of the polygon are numbered in circular order

[2]I.e., computing the fewest *handles* to attach to the sphere so that G can be drawn on the resulting surface without any crossings.

$a_1, b_1, a'_1, b'_1, \ldots, a_g, b_g, a'_g, b'_g$; for $1 \leq k \leq g$, the pairs (a_k, a'_k) and (b_k, b'_k) of sides are identified in opposite direction, meaning that an edge leaving side a_k appears on the corresponding position of side a'_k; see Fig. 3 and Fig. 4a for an example showing K_6 drawn in a fundamental square, which models a drawing on the torus. In such a representation, all vertices lie in the interior of the fundamental polygon and all edges leave the polygon avoiding vertices of the polygon. Alam et al. [2] showed that such a representation can be transformed into a non-simple bundled drawing with g many bundled crossings. It is not clear, however, when such a representation can be transformed into a simple bundled drawing with g bundled crossings, as this transformation can produce drawings with self-intersecting edges and pairs of edges crossing multiple times, e.g., Alam et al. [2, Lemma 1] showed that $\mathrm{bc}(K_6) = 2$ while $\mathrm{bc}'(K_6) = \mathrm{g}(K_6) = 1$.

We now show that computing the bundled crossing number remains NP-hard for simple drawings.

Proof of Theorem 1: Let G' be the graph obtained from G by subdividing each edge $O(|E(G)|^2)$ times. We reduce from the NP-hardness of computing the genus $\mathrm{g}(G)$ of G by showing that $\mathrm{bc}(G') = \mathrm{g}(G)$, with Observation 1 in mind.

Consider the embedding of G onto the genus-$\mathrm{g}(G)$ surface. By a result of Lazarus et al. [24, Theorem 1], we can construct a fundamental polygon representation of the embedding so that its boundary intersects with edges of the graph $O(\mathrm{g}(G)|E(G)|)$ times. Note that each edge piece outside the polygon between the sides of the polygon (a_k, a'_k) intersects each other edge piece outside the polygon between the sides of the polygon (b_k, b'_k) at most once and does not have any other intersection points; see Fig. 3. We then subdivide the edges by adding a vertex to each intersection of an edge with the boundary of the fundamental polygon. This process of subdividing edges ensures that no edge intersects itself or intersects another edge more than once in the corresponding drawing of the graph on the plane; hence, the drawing is simple. Since $\mathrm{g}(G) \leq |E(G)|$, by subdividing edges further whenever necessary, we obtain a drawing of G'. Our subdivisions keep the integrity of all bundled crossings, so $\mathrm{bc}(G') \leq \mathrm{g}(G)$. On the other hand, since subdividing edges does not affect the genus, $\mathrm{g}(G) = \mathrm{g}(G') = \mathrm{bc}'(G') \leq \mathrm{bc}(G')$. □

3 Computing $\mathrm{bc}^{\circ\prime}(G)$ and $\mathrm{bc}^{\circ}(G)$ Is FPT

We now consider circular layouts, where vertices are placed on a circle and edges are routed inside the circle. We note that $\mathrm{bc}^{\circ}(G)$ and $\mathrm{bc}^{\circ\prime}(G)$ can be different.

Observation 2 $\mathrm{bc}^{\circ\prime}(K_{3,3}) = 1$ *but* $\mathrm{bc}^{\circ}(K_{3,3}) > 1$.

Proof: Let $V(K_{3,3}) = \{a, b, c\} \cup \{a', b', c'\}$. A drawing with $\mathrm{bc}^{\circ\prime}(K_{3,3}) = 1$ is obtained by placing the vertices a, a', b, b', c, c' in clockwise order around a circle; see Fig. 2b. If a graph G has $\mathrm{bc}^{\circ}(G) = 1$ then G is planar because we can embed edges for one bundle outside the circle. Hence, $\mathrm{bc}^{\circ}(K_{3,3}) > 1$. □

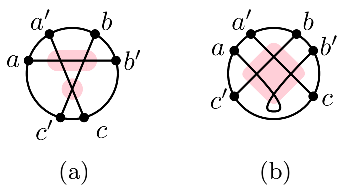


Figure 2: $\mathrm{bc}^\circ(K_{3,3}) \neq \mathrm{bc}^{\circ\prime}(K_{3,3})$; see Observation 2

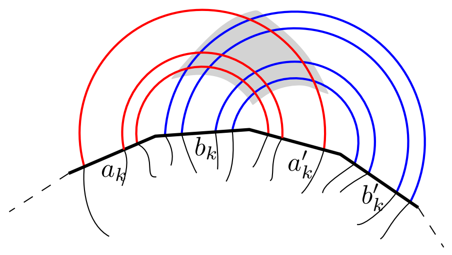
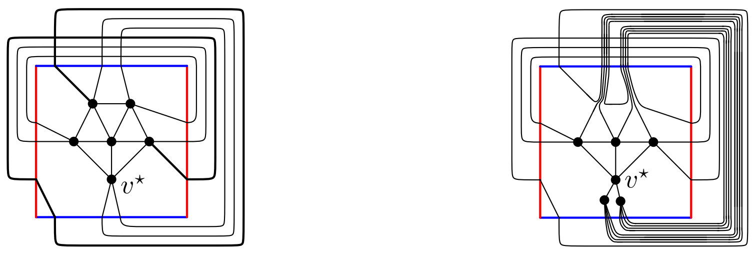


Figure 3: A single bundled crossing outside the fundamental polygon [2, Fig. 3]



(a) K_6 drawn in a fundamental square; the self-intersecting edge is bold [2, Fig. 2]

(b) modifying the representation

Figure 4: Obtaining a circular drawing with k bundled crossings of G from the embedding of $G^\star$ on a surface of genus k.

Similarly to computing $\mathrm{bc}'(G)$, we compute $\mathrm{bc}^{\circ\prime}(G)$ via computing genus. To show this we first prove the following.

Lemma 1 *Given a graph $G = (V, E)$, let $G^\star$ be the graph obtained from G by adding a new vertex $v^\star$ adjacent to every vertex of G. Then* $\mathrm{bc}^{\circ\prime}(G) = \mathrm{g}(G^\star)$.

Proof: Similarly as in [2, Theorem 1], it is easy to see that $\mathrm{bc}^{\circ\prime}(G)$ is an upper bound for the genus of $G^\star$, because, according to Observation 1, we can lift any circular drawing of G onto a surface $\mathcal{S}$ of genus $\mathrm{bc}^{\circ\prime}(G)$ and then we can add $v^\star$ using the outside of the circle. Clearly, this produces a crossing-free drawing of $G^\star$ on the surface $\mathcal{S}$.

It remains to show that given a crossing-free drawing of $G^\star$ on a surface of genus k, we can construct a circular drawing of G with at most k bundled crossings. Consider a drawing of $G^\star$ on a surface $\mathcal{S}$ of genus k.

We use the fundamental polygon representation [2, Theorem 1] to the drawing of $G^\star$ on the surface $\mathcal{S}$ of genus k; see Fig. 4a. Then we modify this representation so that all the neighbors $N(v^\star)$ of $v^\star$ in $G^\star$ are placed in an ϵ-neighborhood of $v^\star$. We now explain the modification in more detail. Consider all the edges incident to $v^\star$ in the representation and drag each neighbor u of $v^\star$ along the edge $uv^\star$ (as illustrated in Fig. 4b) until it reaches the ϵ-neighborhood $N(v^\star)$ of $v^\star$. Since for each $u \in N(v^\star)$ the edges $uw \in E$ with $w \neq v^\star$ are bundled together at the position where u was in the representation and dragged together with u along the edge $uv^\star$, this does not change the number of bundled crossings. Since all the vertices are located on the boundary of the ϵ-neighborhood of $v^\star$ in the modified representation, all the edges between $v^\star$ and $V \setminus v^\star$ are drawn inside the polygon. After removing the vertex $v^\star$ from the representation, we obtain a circular drawing of G with at most k bundled crossings. $\square$

Theorem 4 *Testing whether* $\mathrm{bc}^{\circ\prime}(G) = k$ *can be done in* $2^{k^{O(1)}} n$ *time.*

Proof: By Lemma 1, $\mathrm{bc}^{\circ\prime}(G) = \mathrm{g}(G^\star)$, where $G^\star$ is a graph with a vertex $v^\star$ adjacent to every vertex of G. Applying the $\left(2^{g^{O(1)}} n\right)$-time algorithm for computing genus [22] completes the proof. $\square$

To prove our main result (Theorem 2) we develop an algorithm that tests whether $\mathrm{bc}^\circ(G) = k$ in FPT time with respect to k. Our algorithm is inspired by recent works on circular layouts with at most k crossings [5] and circular layouts where each edge is crossed at most k times [8]. In both of these prior works, it is first observed that the graphs admitting such circular layouts have treewidth $O(k)$, and then algorithms are developed using Courcelle's theorem, which establishes that expressions in MSO_2 logic can be evaluated efficiently. (The basic definition of treewidth is included below; we defer formalizing MSO_2 logic to Section 3.4 where we additionally construct our needed formulas.)

Theorem 5 (Courcelle [10,11]) *For any integer $t \geq 0$ and any MSO_2 formula ψ of length ℓ, an algorithm can be constructed which takes a graph G with n vertices, m edges, and treewidth at most t and decides in $O(f(t,\ell) \cdot (n+m))$ time whether $G \models \psi$ where the function f from this time bound is a computable function of t and ℓ.*

We proceed along the lines of Bannister and Eppstein [5], who used a similar approach to show that edge crossing minimization in a circular layout is in FPT (as mentioned in the introduction). We start by very carefully describing a surface (in the spirit of Observation 1) onto which we will lift our drawing. We will then examine the structure of this surface (and our algorithm) for the case of one bundled crossing and finally for k bundled crossings.

Treewidth The concept of *treewidth* was introduced by Bertelé and Brioschi [6] and then later rediscovered (and popularized) by Robertson and Seymour [31]. A *tree decomposition* of a graph G is a pair (X, T), where T is a tree and $X = \{X_i \mid i \in V(T)\}$ is a family of subsets of $V(G)$, called *bags*, such that (1) for all $v \in V(G)$, the set of nodes $T_v = \{i \in V(T) \mid v \in X_i\}$ induces a non-empty connected subtree of T, and (2) for each edge $uv \in E(G)$ there exists $i \in V(T)$ such that both u and v are in X_i. The maximum of $|X_i| - 1$, $i \in V(T)$, is called the *width* of the tree decomposition. The *treewidth*, $tw(G)$, of a graph G is the minimum width over all tree decompositions of G. For our purposes, an important fact is that every outerplanar graph G has $tw(G) \leq 2$ [26]. An easy observation is that, for any graph G, adding one vertex to G increases its treewidth by at most one, e.g., we simply add this new vertex to every bag of an optimal tree decomposition of G. A direct consequence is that that adding t vertices increases the treewidth by at most t.

3.1 Constructing the Surface Determined by a Bundled Drawing

Consider a bundled circular drawing D. Note that adding parallel edges to the drawing (i.e., making our graph a multi-graph) allows us to assume that every bundled crossing has four distinct frame edges and can be done without modifying the number of bundled crossings; see Fig. 8. Each bundled crossing B defines a Jordan curve made up of the four Jordan arcs $\tilde{e}_1$, $\tilde{e}_2$, $\tilde{e}_3$, $\tilde{e}_4$ in clockwise order taken from its four frame edges $e_1, \ldots, e_4$, respectively, where (e_1, e_3) and (e_2, e_4) frame the two bundles and $e_i = v_{2i-1}v_{2i}$. Similarly to Observation 1, we can construct a surface $\mathcal{S}$ by creating a flat handle (note that this differs from the usual definition of a handle since our flat handles have a boundary) on top of D which connects $\tilde{e}_2$ to $\tilde{e}_4$ and doing so for each bundled crossing. We then lift the drawing D onto $\mathcal{S}$ by rerouting the edges of one of the bundles over its corresponding handle for each bundled crossing B obtaining the lifted drawing $D_{\mathcal{S}}$. To avoid the crossings in $D_{\mathcal{S}}$ of the frame edges that can occur at the foot of the handle of B, we can make the handle a bit wider and add *corner-cuts* (as illustrated in Fig. 5b) to preserve the topology of the surface. Thus, $D_{\mathcal{S}}$ is crossing-free.

We now cut $\mathcal{S}$ into *components* (maximal connected subsets) along the frame edges and corner-cuts of each bundled crossing, resulting in a subdivision Ω of $\mathcal{S}$.

We use D_Ω to denote the sub-drawing of $D_{\mathcal{S}}$ on Ω, i.e., D_Ω is missing the frame edges since these have been cut out. We now consider the components of Ω. Notice that every edge of D_Ω is contained in one component of Ω. In order for a component s of Ω to contain an edge e of D_Ω, s must have both endpoints of e on its boundary. With this in mind we focus on certain components of Ω. Namely, we call a component a *region* if it contains a vertex of G on its boundary. Observe that a crossing in D which does not involve a frame edge corresponds, in D_Ω, to a pair of edges where one goes over a handle and the other goes underneath.

3.2 Recognizing a Graph with One Bundled Crossing

We now discuss how to recognize if an n-vertex graph $G = (V, E)$ can be drawn in a circular layout with one bundled crossing. Consider a bundled circular drawing D of G consisting of one bundled crossing. The bundled crossing consists of two bundles, which are bounded by the set $F = \{e_1, e_2, e_3, e_4\}$ of frame edges. By $V(F)$ we denote the set of vertices incident to frame edges. Via the construction above, we obtain the subdivided surface Ω; see Fig. 5. Let r_1 and r_2 be the regions that are each bounded by a pair of frame edges corresponding to one of the bundles, and let $r_3, \ldots, r_6$ be the regions each bounded by one edge

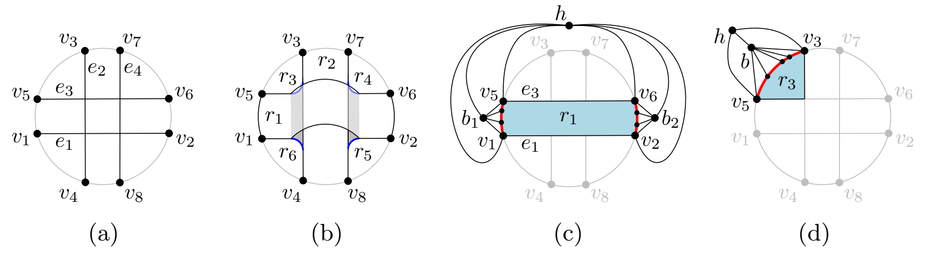


Figure 5: (a) Bundled crossing; (b) regions, corner-cuts in blue; (c),(d) the augmented graphs $G^*_{r_1}$ and $G^*_{r_3}$ consist of the edges of G_{r_1} and G_{r_3} (in the blue regions) as well as augmentation vertices and edges (drawn in black)

from one pair and one from the other pair; see Fig. 5b. These are all the regions of Ω. Since, as mentioned before, each of the non-frame edges of G (i.e., each $e \in E(G) \setminus F$) along with its two endpoints is contained in exactly one of these regions, each component of $G \setminus V(F)$ and each edge connecting it to vertices of $V(F)$ is drawn in D_Ω in some region of Ω. In this sense, for each region r of Ω, we use G_r to denote the subgraph of G induced by the components of $G \setminus V(F)$ contained in r and the edges connecting them to vertices in $V(F)$. Additionally, each vertex of G is either incident to an edge in F (in which case it is on the boundary of at least two regions) or it is on the boundary of exactly one region.

Note that there are two types of regions: those in $\{r_1, r_2\}$ and those in $\{r_3, r_4, r_5, r_6\}$. Consider a region of the first type, say r_1; see Fig. 5b. Observe that G_{r_1} is outerplanar. Moreover, G_{r_1} has a special outerplanar drawing where, on the boundary of r_1, we see (in clockwise order) the frame edge e_1, the vertices mapped to the (v_1, v_5)-arc, the frame edge e_3, and then the vertices mapped to the (v_6, v_2)-arc. We now describe how to augment G_{r_1} to a planar graph $G^*_{r_1}$ where in every planar embedding of $G^*_{r_1}$ the sub-embedding of G_{r_1} has this special outerplanar form[3]. The vertex set of $G^*_{r_1}$ is $V(G_{r_1}) \cup \{h, b_1, b_2\}$ where we call h *hub* vertex and b_1 and b_2 *boundary* vertices (one for each arc of the boundary of r_1 to which vertices can be mapped); see Fig. 5c. The graph $G^*_{r_1}$ has four types of edges; the edges in $E(G_{r_1})$, edges that make h the hub of a wheel whose cycle is $C = (v_6, b_2, v_2, v_1, b_1, v_5, v_6)$, edges from b_1 to the vertices on the (v_1, v_5)-arc, and edges from b_2 to the vertices on the (v_6, v_2)-arc (both including the endpoints). Clearly, we can obtain a planar embedding of $G^*_{r_1}$ by drawing the elements of $G^*_{r_1} \setminus G_{r_1}$ "outside" of the outerplanar drawing of G_{r_1} described before. Moreover, every planar embedding of $G^*_{r_1}$ contains an outerplanar embedding of G_{r_1} that can be drawn in the special form needed to "fit" into r_1, in the sense that all of G_{r_1} lies (or can be put) inside the simple cycle C. (For example, if, say, b_1 is a cut vertex, the component hanging off b_1 can be embedded in the face (h, b_1, v_1, h). But then it can easily be moved into C. Similarly, a component that is incident only to v_5 and v_6 can end up in

[3]This augmentation may sound overly complicated, but is written as to easily generalize to more bundled crossings.

the face (h, v_5, v_6, h), but again, the component can be moved inside C.)

Similarly, for a region of the second type, say r_3, the graph G_{r_3} is outerplanar with a special drawing where all the vertices must be on the (v_3, v_5)-arc of the disk subtended by the two frame edges e_3 and e_2 bounding the region r_3. We augment r_3 similarly as r_1; see Fig. 5d. For the augmented graph $G^*_{r_3}$, we add to G_{r_3} a boundary vertex b neighboring all vertices on the (v_3, v_5)-arc and a hub vertex h adjacent to v_3, b, and v_5. Again, $G^*_{r_3}$ is planar since G_{r_3} is outerplanar. Moreover, as b is adjacent to all vertices of G_{r_3}, in every planar embedding of $G^*_{r_3}$, G_{r_3} is embedded outerplanarly and, since b occurs on one side of the triangle v_3v_5h, the edge v_3v_5 occurs on the boundary of this outerplanar embedding of G_{r_3}. Thus, each planar embedding of $G^*_{r_3}$ provides an outerplanar embedding of G_{r_3} that fits into r_3.

Note that each G_{r_i} fits into r_i because its augmented graph $G^*_{r_i}$ is planar ($\star$). Moreover, as outerplanar graphs have treewidth at most two [26], each graph G_r is outerplanar, and adding the (up to) eight frame vertices raises the treewidth by at most 8, we see that the treewidth of G is at most 10. Namely, in order for G to have $\mathrm{bc}^\circ(G) = 1$, it must have treewidth at most 10 (and this can be checked in linear time using an algorithm of Bodlaender [7]).

To sum up, G has a circular drawing D with at most one bundled crossing because it has treewidth at most 10 and there exist (i) $\beta \leq 4$ frame edges $e_1, e_2, \ldots, e_\beta$ (this set is denoted F) and $v_1, \ldots, v_\xi$ frame vertices (this set is denoted V_F), (ii) a particular circular drawing D_F of frame edges, (iii) the drawing of the one bundled crossing B, and (iv) $\gamma \leq 6$ corresponding regions $r_1, \ldots, r_\gamma$ of the subdivided surface Ω so that the following properties hold. (Note that the frame vertices partition the boundary of the disk underlying Ω into $\eta \leq 8$ (possibly degenerate) arcs $p_1, \ldots, p_\eta$ where each such p_j is contained in a unique region r_{i_j} of Ω. Let $V_0(r_i)$ be the frame vertices incident to region r_i.)

1. $E(G)$ is partitioned into $E_0, E_1, \ldots, E_\gamma$, where $E_0 = \{f_1, \ldots, f_\beta\}$.

2. $V(G)$ is partitioned into $V_0, V_1, \ldots, V_\eta$, where $V_0 = \{w_1, \ldots, w_\xi\}$.

3. The mapping $w_i \leftrightarrow v_i$ and $f_i \leftrightarrow e_i$ defines an isomorphism between the subgraph of G formed by (V_0, E_0) and the graph (V_F, F).

4. For each $v \in V_0$ and each edge e incident to v, exactly one of the following conditions holds: (i) $e \in E_0$, or (ii) $e \in E_i$ and v is on the boundary of r_i.

5. For each $v \in V_j, j \neq 0$, all edges incident to v belong to E_{i_j}.

6. For each region r_i, let G_{r_i} be the graph $(V_0(r_i) \cup \bigcup_{j\,:\,i_j = i} V_j, E_i)$ (i.e., the subgraph that is to be drawn in r_i), and let $G^*_{r_i}$ be the corresponding augmented graph (i.e., as in $\star$ above). Each $G^*_{r_i}$ is planar.

We now describe the algorithm that tests whether a given graph G admits a simple circular drawing with one bundled crossing. First we check that the treewidth of G is at most 10. We then enumerate drawings of up to four edges in the circle. For the drawing D_F that is valid for the set F of frame edges of one bundled crossing, we define our surface and its regions (which makes the augmentation well-defined). We have intentionally phrased these properties so that it is clear that they are expressible in MSO_2 (see Section 3.4). The only property that is not obviously expressible is the planarity of $G^*_{r_i}$. To this end, recall that planarity is characterized by two forbidden minors (i.e., K_5 and $K_{3,3}$) and that, for every fixed graph H, there is an MSO formula $\textsc{minor}_H$ so that for all graphs G, it holds that $G \models \textsc{minor}_H$ if and only if G contains H as a minor [11, Corollary 1.14]. Additionally, each $G^*_{r_i}$ can be expressed as an *MSO-transduction*[4] of G and our variables (our transduction can be thought of as a kind of 2-copying transduction). Thus, by [11, Theorem 7.10] using the transduction and the MSO formula testing planarity, we can construct an MSO_2 formula ι so that when $G \models \iota$, $G^*_{r_i}$ is planar for every i. Therefore, Properties 1–6 can be expressed as an MSO_2 formula ψ and, by Courcelle's theorem, there is a computable function f such that we can test (in $O(f(\psi, t)n)$ time) whether $G \models \psi$ for an input graph G of treewidth at most t. Thus, since our graph has treewidth at most 10, applying Courcelle's theorem completes our algorihtm.

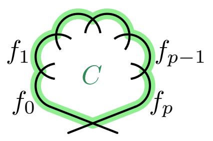
3.3 Recognizing a Graph with k Bundled Crossings

We now generalize the above approach to k bundled crossings. In a drawing D of G together with a solution consisting of k bundled crossings, there are $2k$ bundles making (up to) $4k$ frame edges F. As described above, these bundled crossings provide a surface $\mathcal{S}$, its subdivision Ω, and the corresponding set of regions. The key ingredient above was that every region r contained an outerplane graph G_r. However, that is now non-trivial as our regions can go over and under many handles. To show this property, we first consider the following two partial drawings $D_\mathrm{A}(p)$ and $D_\mathrm{B}(p)$ of a matching with $p+1$ edges $f_0, f_1, \ldots, f_p$ (see Fig. 6) such that

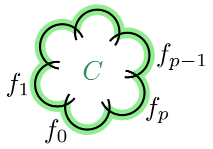
- edge f_i crosses only $f_{i-1 \bmod p+1}$ and $f_{i+1 \bmod p+1}$ for $i = 0, \ldots, p$;
- the endpoints of each edge f_i, $i = 1, \ldots, p-1$, are inside the closed curve C formed by the crossing points and the edge-pieces between these crossing points;
- only one endpoint of f_0, and only one endpoint of f_p are contained in C in the drawing $D_\mathrm{A}(p)$;
- both endpoints of f_0 and f_p are contained in C in the drawing $D_\mathrm{B}(p)$.

Note that the partial drawings $D_\mathrm{A}(p)$ and $D_\mathrm{B}(p)$ differ only in how the last edge is drawn with respect to the first one. Arroyo et al. [4, Theorem 1.2] showed that such partial drawings are obstructions for pseudolinearity, that is, they cannot be part of any pseudoline arrangement. Therefore, neither of these partial drawings can be *completed* to a simple circular drawing, that is, the

[4]For the formalities of transductions, see the book of Courcelle and Engelfriet [11, Section 1.7.1, and Definitions 7.6 and 7.25].



(a) $D_{\mathrm{A}}(p)$



(b) $D_{\mathrm{B}}(p)$

Figure 6: The two types of partial drawings (for $p = 6$) and the closed curve C (light green) that they induce.

endpoints of the edges cannot be extended so that they lie on a circle which contains the drawing. We highlight this fact in the following lemma.

Lemma 2 ([4]) *For a matching with $p+1$ edges $f_0, f_1, \ldots, f_p$, neither the partial drawing $D_{\mathrm{A}}(p)$ nor the partial drawing $D_{\mathrm{B}}(p)$ can be completed to a simple circular drawing.*

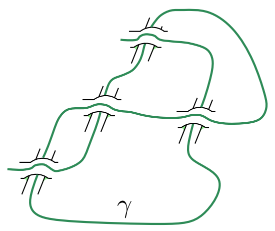
Using this lemma, we can now prove the following statement.

Lemma 3 *Let r be a region of the surface subdivision Ω, and let r' be its projection onto the plane. Then both r' and r are topological disks, that is homeomorphic to a disk (with the boundary). Moreover, the projection map is injective.*

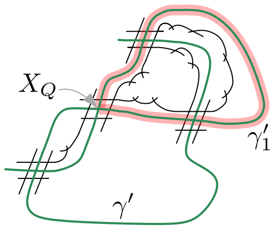
Proof: Note that the boundary of r is formed by pieces of frame edges that were lifted on the surface $\mathcal{S}$ as described above and by additional corner-cuts as illustrated in Fig. 5b in blue. This means that the boundary of r is a (closed) Jordan curve since we have only finitely many crossings in G. Then, we show that r does not include part of both a handle and its *undertunnel*, that is, the part of the surface over which the handle was built. This guarantees that the projection of r onto the plane is injective, and thus the boundary of r' is a Jordan curve. We will also show that r does not include holes. Then we can conclude that r' is homeomorphic to a disk using the Jordan–Schoenflies theorem, which says that for any closed Jordan curve there is a homeomorphism of the plane that maps the curve to the unit circle.

Suppose now, for a contradiction, that there are bundled crossings for which r contains both the handle and its undertunnel; see Fig. 7a. Then there exists a non-intersecting Jordan arc $\gamma \subset r$ going over and under some of these handles. Consider the orthogonal projection γ' of γ on the disk of the drawing D (see Fig. 7b) and notice that it self-intersects where it went over and under some handle in r. Choose the piece γ'_1 of γ' separated by the self-intersection point X_Q, corresponding to some bundled crossing Q, such that γ'_1 starts and ends in X_Q and no intersection point (except X_Q) is met twice when walking along γ'_1 once; see Fig. 7b.

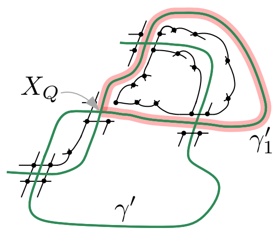
Let P be the planarization of the projected drawing, and let P' be a copy of P without the edges that intersect γ'_1. Consider the edges of P' in the interior of the closed curve γ'_1 (see Fig. 7c) that can be *seen* from γ'_1, that is, for each



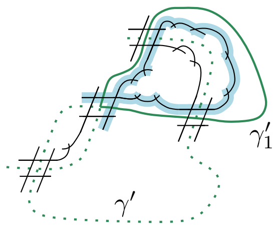
(a) A region r with handles and undertunnels corresponding to the same bundled crossings and a curve γ that goes over and under such handles;



(b) projection of the region r and projection γ' of the curve γ on the drawing D in the plane. The curve γ'_1 (light red) is a part of γ' that starts and ends in X_Q.



(c) the planarization P';



(d) the profile of the curve γ'_1 (light blue). These edges form a partial drawing $D_{\mathrm{A}}(p)$.

Figure 7: Illustration to the proof of Lemma 3.

of such edges, we can draw a curve β from some point of γ'_1 so that β does not intersect γ'_1 and any other edge of P'. We call the edges of the drawing D that contain the edges of the planarization P' seen from γ'_1 the *profile* of γ'_1; see Fig. 7d. These edges form a partial drawing $D_{\mathrm{A}}(p)$ for some $p > 0$; see Fig. 6a. According to Lemma 2, however, such a partial drawing cannot be completed to a valid simple circular drawing; contradiction.

As for holes, it is easy to see that if r had a hole, the profile of any curve around this hole would yield a partial drawing $D_{\mathrm{B}}(p)$ for some $p > 0$; see Fig. 6b. Again, according to Lemma 2, such a partial drawing cannot be completed to a valid simple circular drawing; contradiction.

Note that, since r' is a topological disk, its lifting r is also a topological disk. □

In particular, since our projection is injective, a drawing on r can be regarded as a drawing on r' and vice versa.

The next lemma concerning treewidth is a direct consequence of Lemma 3.

Lemma 4 *If a graph G admits a circular layout with k bundled crossings then its treewidth is at most $8k + 2$.*

Proof: If the graph G can be drawn in a circular layout with k bundled crossings then there exist at most $4k$ frame edges. According to Lemma 3, the

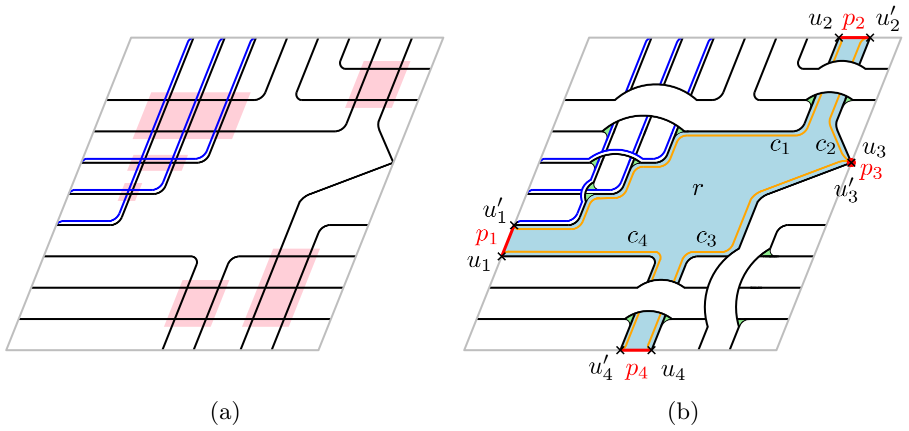


Figure 8: (a) A bundled drawing D with six bundled crossings (pink); parallel (blue) edges can be inserted to avoid degenerate bundled crossings; (b) the corresponding surface of genus 6; the components of the surface that are not regions are marked in green; the region r (light blue) has a boundary consisting of the arcs of the disk (red) and the arcs c_1, c_2, c_3, and c_4 (traced in orange).

removal of their endpoints breaks up the graph into outerplanar components. The treewidth of an outerplanar graph is at most two [26]. Moreover, adding a vertex to a graph raises its treewidth by at most one. Thus, since deleting at most $8k$ frame vertices leaves behind an outerplanar graph, G has treewidth at most $8k + 2$. □

We now prove Theorem 2, which says that deciding whether $\mathrm{bc}^\circ(G) \leq k$ is FPT in k.

Proof of Theorem 2: We use Lemma 3 and extend the algorithm of Section 3.2.

Suppose that G has a circular drawing D with at most k bundled crossings. Then D contains a set F of (up to) $4k$ frame edges of these bundled crossings. As discussed before, F together with D defines a subdivided topological surface Ω containing a set R of regions. As in the case of one bundled crossing, each edge of G not in F is contained in exactly one such region, and each vertex of G either is incident to an edge in F (in which case it belongs to at least two regions) or belongs to exactly one region.

Throughout the proof we refer the reader to Fig. 8 for an example. By Lemma 3, each region r in R is a topological disk. Therefore, the graph G_r whose vertices lie on the boundary of r and whose edges lie in the interior of r is outerplane with respect to the given order of the vertices along the boundary. This boundary consists, in clockwise order, of arcs $p_1, \ldots, p_\alpha$ of the outer boundary of $\mathcal{S}$ (marked in red in Fig. 8b) and Jordan arcs $c_1, \ldots, c_\alpha$ (traced in orange in Fig. 8b), each of which connects two consecutive arcs of $\mathcal{S}$. For $i \in \{1, \ldots, \alpha\}$, let u_i and u'_i be the endpoints of p_i, in clockwise order. The

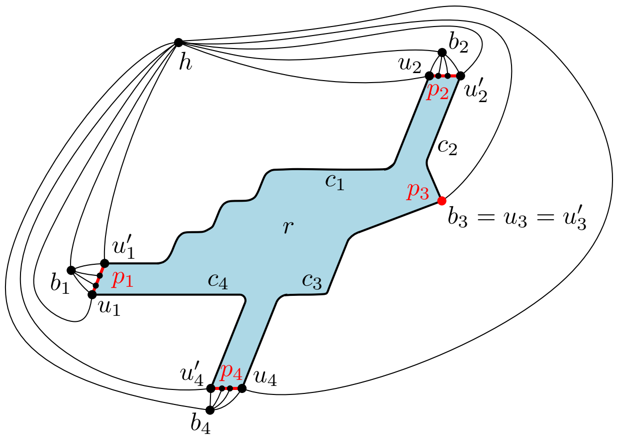


Figure 9: The augmented graph G_r^* for a complex region r. The arc p_3 is degenerate.

arc p_i can degenerate to a single point; then $u_i = u'_i$; see Fig. 9. So u'_i and u_{i+1} (where $u_{\alpha+1}$ is u_1) are the endpoints of c_i. No vertex of G_r lies in the interior of c_i.

We now describe G_r^*. First, we add a hub vertex h. Then, for each $i \in \{1, \ldots, \alpha\}$, if u'_i and u_{i+1} (where $u_{\alpha+1}$ is u_1) are not adjacent, we add an edge between them. If the arc p_i is non-degenerate, we add a boundary vertex b_i adjacent to all vertices on p_i (including u_i and u'_i) and make h adjacent to u_i, b_i, and u'_i. Otherwise, we make h adjacent to $u_i = u'_i$ and identify b_i with u_i and u'_i. The reason for this identification is technical; it allows us to iterate over all (degenerate or non-degenerate) arcs and address their boundary vertices; see Section 3.4 (page 20).

Observe that the resulting graph G_r^* is planar due to the special outerplanar drawing of G_r in r. Moreover, in every planar embedding of G_r^*, there is an outerplanar embedding of G_r where the cyclic order of the arcs c_i and the sets of vertices mapped to the p_i's match their cyclic order in r, implying that G_r fits into r. This is due to the fact that the simple cycle C' around h must be embedded planarly, with all of G_r inside (with the possible and easy-to-fix exceptions described in Section 3.2 concerning the cycle C there). Then the order of the vertices in an outerplanar embedding of G_r is the order of the vertices incident to $b_1, \ldots, b_\alpha$ in a planar embedding of G_r^*. So the planarity of G_r^* guarantees that G_r fits into r as needed.

The reason why G has a circular drawing D with at most k bundled crossings is that there is a β-edge k-bundled crossing drawing D_F (of the graph formed by F), whose corresponding surface $\mathcal{S}$ consists of regions $r_1, \ldots, r_\gamma$ (note: $\gamma \leq 2\beta \leq 8k$) so that Properties 1–6 hold.

Our algorithm first checks that the treewidth of G is at most $8k + 2$. Recall that this can be done in linear time (FPT in k) [7]. The algorithm then enumerates all possible simple drawings of at most $4k$ edges in the circle[5]. For each drawing, it further enumerates the possible ways to form at most k bundled crossings so that every edge is a frame edge of at least one bundled crossing. Then, for each such bundled drawing D_F, we build an MSO_2 formula φ (see Section 3.4) to express Properties 1–6. Finally, since G has treewidth at most $8k+2$, we can apply Courcelle's theorem on (G, φ). □

3.4 MSO$_2$: Definitions and Our Formula for a Specific Layout of the Frame Edges

This subsection serves two purposes. First, we define the class MSO_2 of logical formulas as needed for our application of Courcelle's theorem; see Theorem 5. Second, we describe how to express the needed condition of our algorithm (as given by Properties 1–6) in this logic.

Defining MSO$_2$. The class of formulas expressible in MSO_2 is defined as follows; see also the textbook of Courcelle and Engelfriet [11] for more background.

Extended Monadic Second-Order Logic (MSO_2) is a subset of *second-order logic* that can be used to express certain graph properties. It is built from the following primitives:

- variables for vertices, edges, sets of vertices, and sets of edges;
- binary relations for: equality ($=$), membership in a set ($\in$), subset of a set ($\subseteq$), and edge–vertex incidence (I);
- standard propositional logic operators: $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$;
- standard quantifiers ($\forall, \exists$) which can be applied to all types of variables.

Note that, if we drop the "$_2$" then we have *Monadic Second-Order Logic* (MSO) where the only difference is that we are now not allowed to quantify over edge sets. Additionally, the convention for MSO formulas is to use a binary *adjacency function (adj$_G$)* instead of the incidence function as in the definition of MSO_2 above.

For a graph G and an MSO_2 (or MSO) formula ψ, we use $G \models \psi$ to indicate that ψ can be satisfied by G in the obvious way. Note that we will use an additional tool in the construction of our formula below, namely, that of *MSO-transductions.* Essentially, an *L-transduction* is just the name for the operation of constructing the model of one graph/structure from the model of another graph/structure in the language of the logic L. A rigorous treatment on transductions is given in the book of Courcelle and Engelfriet [11, Section 1.7.1, and Definitions 7.6 and 7.25]. For example, an MSO-transduction that constructs

[5] i.e., at most $4k$ curves extending to infinity in both directions where each pair of curves cross at most once. The number of such drawings is proportional to k, and efficient enumeration has been done for the case when every pair of curves cross exactly once [14].

a graph G' (modelled for MSO, using a vertex set $V(G')$ and adjacency function $adj_{G'}$) by adding a universal vertex x to a given graph G (modelled for MSO$_2$, using a vertex set $V(G)$, edge set $E(G)$, and incidence function I_G) can be written as follows:

$$\begin{aligned} V(G') &:= \{x\} \cup V(G) \\ adj_{G'}(u,v) &:= (u \neq v) \wedge \\ &\quad \Big(\big((\exists e \in E(G))\ (I_G(e,v) \wedge I_G(e,u))\big) \vee (x = u) \vee (x = v)\Big). \end{aligned}$$

We will describe our augmented graphs (from Property 6) as an MSO-transduction (also whose target is modelled for MSO and whose original graph is modelled for MSO$_2$) and this will allow us (via [11, Theorem 7.10]) to have an MSO$_2$ formula to implicitly check the planarity of our augmented graphs inline within our (main) MSO$_2$ formula (where our formula is applied only to the graph prior to augmentation).

The formula for D_F. We now construct an MSO$_2$ formula to express the following problem:

- Given a graph $G = (V,E)$ and a simple circular drawing D_F with k bundled crossings so that $F = \{e_1, \ldots, e_\beta\}$ is the set of frame edges (and D_F has no other edges) and $V_F = \{v_1, \ldots, v_\xi\}$ is the set of frame vertices (and D_F has no other vertices);
- determine whether G has a simple circular drawing with k bundled crossings so that the frame edges and vertices occur as in D_F.

This is based on Properties 1–6 on page 11: we express them as MSO$_2$ formulas.

Properties 1 and 2 simply state that a set of elements is partitioned into a certain number of disjoint subsets. We use a formula stated by Bannister and Eppstein [5] to express this in MSO$_2$. For example, partitioning of a set E into disjoint subsets $E_0, E_1, \ldots, E_\gamma$ can be done as follows:

$$\textsc{Partition}(E; E_0, \ldots, E_\gamma) = (\forall e \in E)\Big[\big(\bigvee_{i=0}^{\gamma} e \in E_i\big) \wedge \big(\bigwedge_{i \neq j} \neg(e \in E_i \wedge e \in E_j)\big)\Big].$$

We will additionally use the following formula to state that a vertex set V' is the set of endpoints of an edge set E':

$$\textsc{Incident}(V', E') \;=\; (\forall e \in E')\ (\forall v \in V(G))\ [I(e,v) \Leftrightarrow v \in V'].$$

We now turn to the properties more specific to our fixed drawing D_F with $\beta \leq 4k$ frame edges $F = \{e_1, e_2, \ldots, e_\beta\}$ whose endpoints form the set $V(F) = \{v_1, v_2, \ldots, v_\xi\}$, where $\xi \leq 2\beta$. As discussed in Section 3.1 and Lemma 3, this drawing induces a corresponding set of regions $r_1, \ldots, r_\gamma$.

Property 3 ensures that certain edges $E_0 = \{f_1, f_2, \ldots, f_\beta\}$ and their endpoints $V_0 = \{w_1, w_2, \ldots, w_\xi\}$ of the graph G induce a graph isomorphic to $(V(F), F)$. This can be modeled by the following formula.

$$\begin{aligned}\theta_3(V_0, E_0) \;=\; &\big(\forall i, j \in \{1, 2, \ldots, \xi\}\big)\\ &\Big[\big((\exists f \in E_0)\ I(e, w_i) \wedge I(f, w_j)\big) \Leftrightarrow \big((\exists e \in F)\ I(e, v_i) \wedge I(e, v_j)\big)\Big]\end{aligned}$$

To express Properties 4 about the adjacencies of the frame vertices, we introduce the following piece of notation. For each vertex $v_i \in V(F)$ with $i \in \{1, 2, \ldots, \xi\}$, we denote by $\sigma(i)$ the set of indices of the regions incident to v_i in the drawing D_F. For example, in the case of one bundled crossing (see Fig. 5), $\sigma(1) = \{1, 6\}$. Then Property 4 can be expressed in MSO$_2$ as follows:

$$\begin{aligned}\theta_4(V_0, E_0) \;=\; &\big(\forall i \in \{1, 2, \ldots, \xi\}\big)\ (\forall e \in E)\\ &\Big[I(e, w_i) \Rightarrow \big[e \in E_0 \vee (\exists j \in \sigma(i))\,[e \in E_j]\big]\Big].\end{aligned}$$

Property 5 expresses that, for each non-frame vertex $v \in V_j$, all edges incident to v are contained in E_{i_j} (recall that i_j is the index of the region containing the set V_j of non-frame vertices and that E_{i_j} is the set of non-frame edges of this region):

$$\theta_5(V_1, \ldots, V_\eta) \;=\; \big(\forall j \in \{1, 2, \ldots, \eta\}\big)\,\big(\forall v \in V_j\big)\,\big(\forall e \in E\big)\,\big[I(e, v) \Rightarrow e \in E_{i_j}\big]$$

Finally, we turn to Property 6. First, note that testing planarity of a graph G can be expressed as follows where the formula for $\textsc{Minor}_H(G)$ does not need edge set quantification (i.e., it is in MSO) [11, Corollaries 1.14 and 1.15]:

$$\textsc{Planar}(G) \;=\; \neg\textsc{Minor}_{K_5}(G) \wedge \neg\textsc{Minor}_{K_{3,3}}(G).$$

Now, we describe the MSO-transduction[6] τ_i of G to $G^*_{r_i}$ (for each region r_i; see Section 3.3) subject to the variables $w_1, \ldots, w_\xi$, $V_1, \ldots, V_\eta$, $f_1, \ldots, f_\beta$, $E_1, \ldots, E_\gamma$. Note that in our transduction, the input uses the format allowing for edge set quantification (i.e., where we have the objects $V \cup E$ and the binary incidence function I), but our output involves the format without edge set quantifications (i.e., where we have the objects V and the binary adjacency function adj). Let $j_1, \ldots, j_\zeta$ be the indices of the frame vertices incident to region r_i and suppose these are ordered cyclically as in D_F. Further, let $V_{l_1}, \ldots, V_{l_\alpha}$ be the sets corresponding to the arcs of the boundary of r_i (in order). With this notation, we can now set up the transduction τ_i which describes our graph $G^*_{r_i}$ in terms of our variables (note that in the statement of [11, Theorem 7.10] our variables are the *parameters*). Note that the symbols $h, b_1, b_2, \ldots, b_\alpha$ are new objects that are added in the construction (namely, the hub and boundary vertices of $G^*_{r_i}$). Further, let C be the cycle of the wheel, that is, $V(C) = \{v_{j_1}, \ldots, v_{j_\zeta}, b_1, \ldots, b_\alpha\}$.

[6]Recall that, a *transduction* is essentially just the name for the operation of constructing the model of one graph/structure from the model of another graph/structure in the language of MSO.

For each vertex $x \in V(C)$, let $N_C(x)$ be the set consisting of the two neighbors of x in C.

Now we can describe the transduction τ_i as follows.

$$\begin{aligned}
V(G^*_{r_i}) :=\ & \{h\} \cup V(C) \cup \bigcup_{j=1}^{\alpha} V_{l_j} \\
adj_{G^*_{r_i}}(u,v) :=& (u \neq v) \wedge \\
& \Big(\big((\exists e \in E_i)\ (I(e,v) \wedge I(e,u))\big) \\
& \vee \big((h = u) \wedge (v \in V(C))\big) \vee \big((h = v) \wedge (u \in V(C))\big) \\
& \vee \left(\bigvee_{j=1}^{\alpha} u = b_j \wedge v \in V_{l_j} \right) \vee \left(\bigvee_{j=1}^{\alpha} v = b_j \wedge u \in V_{l_j} \right) \\
& \vee \big((u \in V(C)) \wedge (v \in N_C(u))\big) \\
& \vee \big((v \in V(C)) \wedge (u \in N_C(v))\big) \Big).
\end{aligned}$$

With this transduction τ_i and the expression $\textsc{Planar}(G)$, we can now apply [11, Theorem 7.10] to obtain the MSO$_2$ formula ι_i which, when applied to G (together with our variables), allows us to express that $G^*_{r_i}$ is planar. Namely, by taking the conjunction of all of these ι_i, we obtain the needed MSO$_2$ formula ι (which can be applied to G and our variables) to express that all of the $G^*_{r_i}$'s are planar.

Now we construct the MSO$_2$ formula corresponding to Properties 1–6. The formula depends on the drawing D_F of the set of frame edges F.

$$\begin{aligned}
\textsc{realizable}_{D_F}(G) \ \equiv\ & \\
& (\exists f_1, \dots, f_\beta, E_0, E_1, \dots, E_\gamma, w_1, w_2, \dots, w_\xi, V_0, V_1, \dots, V_\eta) \\
& \Big[E_0 = \{f_1, \dots, f_\beta\} \wedge V_0 = \{w_1, w_2, \dots, w_\xi\} \\
& \wedge\ \textsc{Partition}(E; E_0, E_1, \dots, E_\gamma) \\
& \wedge\ \textsc{Partition}(V; V_0, V_1, \dots, V_\eta) \\
& \wedge\ \textsc{Incident}(V_0, E_0) \\
& \wedge\ \theta_3(V_0, E_0)\ \wedge\ \theta_4(V_0, E_0)\ \wedge\ \theta_5(V_1, \dots, V_\eta) \\
& \wedge\ \iota(f_1, \dots, f_\beta, E_1, \dots, E_\gamma, w_1, w_2, \dots, w_\xi, V_1, \dots, V_\eta) \Big].
\end{aligned}$$

4 Bundling a Drawing

We now establish the following parameterized results for bundling a given drawing.

Theorem 6 *Let G be a graph with n vertices and $m \geq n$ edges, and let D be a drawing of G.*

(a) If D has c crossings, we can test whether $\mathrm{bc}'(G, D) \leq k$ *in* $c^{O(\sqrt{k})} + O(m + c)$ *time.*

(b) If D is simple, we can test whether $\mathrm{bc}(G, D) \leq k$ *in* $m^{O(\sqrt{k})} + O(m)$ *time.*

(c) If D is simple and circular, testing whether $\mathrm{bc}^\circ(G, D) \leq k$ *is FPT in k; it takes* $2^{O(\sqrt{k}\log k)} + O(m)$ *time.*

(d) For a permutation π of $V(G)$, testing whether $\mathrm{bc}^\circ(G, \pi) \leq k$ *is FPT in k; it takes* $2^{O(k^2)} + O(m)$ *time.*

To prove this theorem, we will examine the number of combinatorially different bundled crossings we can make in our given fixed drawing D. Namely, let $\mathcal{B}(D)$ be the entire family of subsets of crossings in D such that each subset corresponds to a bundled crossing in D, that is, for each bundled crossing, the subset S of the crossings in D contained in it is an element of $\mathcal{B}(D)$. We show that $|\mathcal{B}(D)| \in O(c^4)$ where c is the number of crossings in D; see Lemma 6. For the case when D is simple, we show that $|\mathcal{B}(D)| \in O(m^4)$ where m is the number of edges in D; see Lemma 5.

Note that each element of $\mathcal{B}(D)$ forms a connected subgraph in the planarization of the drawing. So, by finding k such connected subgraphs that are pairwise disjoint and together cover the crossings of D, we can bundle the drawing to have at most k crossings. Marx and Pilipczuk [25] studied exactly this type of disjoint covering problem. Their result is as follows.

Theorem 7 ([25, Theorem 1.3]) *Let G be a planar graph, let B be a family of connected vertex sets in G, let $C \subseteq V(G)$ be a set of vertices, and let k be an integer. In time $|B|^{O(\sqrt{k})} n^{O(1)}$, we can find a set S of at most k pairwise disjoint objects in B that maximizes the number of vertices of C in the union of the vertex sets in S.*

We use Theorem 7 and an algorithm of Alam et al. [2] to prove Theorem 6.

Proof of Theorem 6: (a) Consider a drawing D of G with c crossings, and let $B = \mathcal{B}(D)$. By Lemma 6, $|B| \in O(c^4)$. Let D' be the plane graph obtained from D by creating a vertex at each crossing point in D (note that D' does not contain the vertices of G), and connecting two such vertices if they are consecutive along an edge in D. Clearly, each element in B forms a connected subgraph of D'. Thus, applying Theorem 7 with $G = D', B = \mathcal{B}(D), C = V(D')$ establishes (a).

(b) This follows as in (a). Namely, since D is simple, by Lemma 5, $|\mathcal{B}| \in O(m^4)$. This establishes (b).

(c) Alam et al. [2] showed that testing $\mathrm{bc}^\circ(G, \pi) \leq k$ can be kernelized down to an instance with at most $16k$ edges (or report that (G, π) is a no-instance) in $O(m)$ time. This also applies to a given simple circular drawing. Thus, by applying their kernelization and using (b) with $m \leq 16k$, we establish (c).

(d) Here, we use (b) to improve the FPT algorithm of Alam et al. [2] for determining whether $\mathrm{bc}^\circ(G, \pi) \leq k$. (Their algorithm runs in time $k^{O(k^2)} + O(m)$ and proceeds in three stages). First, it applies a kernelization step to obtain a graph with at most $16k$ edges in $O(m)$ time; second, it enumerates all possible $O\big(2^{0.657k^2}\big)$ weak pseudoline arrangements of $16k$ pseudolines [14]; and third, for each such weak pseudoline arrangement it partitions the crossings into the minimum number of bundled crossings by exhaustive search in time $O\big(k^{128k^2}\big)$. For this last step, we apply (b) instead (now with $m \leq 16k$), leading to $(16k)^{O(\sqrt{k})} = 2^{O(\sqrt{k}\log k)}$ time, and $2^{O(k^2)} + O(m)$ time in total. □

Note that since the number of crossings in a non-simple drawing is not bounded by a function of the number of edges m in the drawing, we do not obtain an analogous result for the circular layout as in Theorem 6(c) by using the kernelization technique of Alam et al. [2]. On the other hand, we present a $2^{O(m)}$-time algorithm for not necessarily simple drawings in circular layouts in the context of storyline visualization; see Section 5.

We now discuss the size of the family $\mathcal{B}(D)$ for some drawing D. Note that each bundled crossing in D involves two pairs of frame arcs, and, conversely, two pairs of frame arcs can determine at most one bundled crossing. We show that if D is simple, then this is also true for frame edges, that is, two pairs of frame edges can determine at most one bundled crossing; see Lemma 5. This allows us to bound the number of distinct bundled crossings by the number of edges from above, and thus, the size of the family $\mathcal{B}(D)$.

Lemma 5 *Let D be a simple drawing D with m edges. Each bundled crossing determines at most two pairs of frame edges, and, conversely, two pairs of frame edges can determine at most one bundled crossing. In particular, $|\mathcal{B}(D)| \leq m^4$.*

Proof: Consider two bundles that form a given bundled crossing. Each bundle has at most two not necessarily distinct frame edges. For the reverse direction, consider a bundled crossing B and let (e_2, e_4), (e_1, e_3) be the two pairs of frame edges each corresponding to a bundle. Let c_{ij}, for $i = 1, 3$, $j = 2, 4$ be the frame crossing of e_i and e_j (if it exists). There are three cases how these two pairs can determine a bundled crossing: (a) $e_1 = e_3$ and $e_2 = e_4$: then B is a single crossing, clearly there cannot be another bundled crossing determined by the same pairs; (b) $e_1 = e_3$ and $e_2 \neq e_4$ (the case where $e_1 \neq e_3$ and $e_2 = e_4$ is symmetric): then B consists of all crossings on e_1 between c_{12} and c_{14}, since e_1 and e_2 can cross e_1 at most once, there cannot be another bundled crossing; or (c) the edges are pairwise different, then all the crossings c_{ij} exist and are distinct and any other bundled crossing would imply that for some fixed i and j the crossings c_{ij} occurred twice, which is impossible in a simple drawing. □

In the case of a not necessarily simple drawing D, the number c of crossings cannot be bounded in terms of the number m of edges. But if we define the size of an instance in terms of the number of crossings, it is easy to see that $\mathcal{B}(D)$ is of size polynomial in c.

Lemma 6 *For any not necessarily simple drawing D with c crossings $|\mathcal{B}(D)| \in O(c^4)$.*

Proof: Every bundled crossing can be determined by two pairs of frame arcs. Since each crossing can be incident to at most four arcs, four crossings can determine at most 4^4 different bundled crossings. Therefore, the total number of bundled crossings is polynomially bounded by the number of crossings, namely $|\mathcal{B}(D)| \leq 16c^4$. □

5 Bundling Storyline Visualisations Is FPT

For our purposes, a storyline drawing D is a set of m x-monotone curves. Such curves cannot self-intersect, but a pair of curves is allowed to intersect each other multiple times; we only forbid the existence of digonal faces, that is, two curves intersecting each other twice in a row. Finally, we assume here that all curves start on distinct points of a vertical line v_{left} and end on distinct points of a vertical line v_{right}. This is common in storyline visualizations, but this restriction can be dropped with additional care. We prove the following.

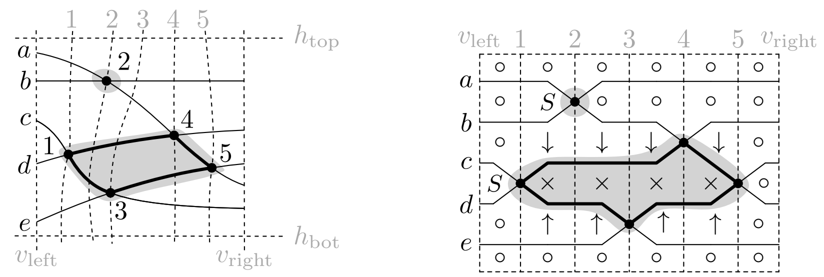
Theorem 8 *Given a storyline drawing D with m characters and c crossings, $\text{bc}^{\text{s}}(D)$ can be computed in $O(\varphi^{2m}mc) \subset O(2.62^m mc)$ time, where φ is the golden ratio. This runtime is fixed parameter tractable in m. An optimal bundling can be constructed in the same time.*

Recall that $I(D)$ is the set of crossings. Each curve, being x-monotone, gives a left-to-right order of its incident crossings. These orders give a partial order on $I(D)$. Let π be an arbitrary linear extension of these partial orders, which can be found in polynomial time given D. Then we subdivide D into columns according to π: see Definition 2 and Fig. 10. We call a face of this subdivision a *cell*. In this section we define a way to label the cells to describe any bundling of the drawing. The algorithm for Theorem 8 is based on dynamic programming over such labelings.

Definition 2 *A subdivision $\mathcal{S}(D)$ consists of the drawing D together with: horizontal lines h_{top} and h_{bot} above and below all curves; vertical lines v_{left} and v_{right} going through the left and right endpoints of the curves, respectively; and a set of c y-monotone curves with the following properties:*

- *for each crossing X in $I(D)$, there is a unique curve going through X,*
- *each curve crosses h_{top}, h_{bot} and all curves of D,*
- *the curves do not intersect each other, and they are totally ordered from left to right according to π.*

See Fig. 10a for an example of a drawing D with its subdivision $\mathcal{S}(D)$. Let $\mathcal{C}$ be the set of bounded faces of $\mathcal{S}(D)$; we call the elements of $\mathcal{C}$ the cells in order to distinguish them from the faces of D. A drawing of $\mathcal{S}(D)$ in Fig. 10b helps to understand its structure. (This drawing is stretched similarly to a



(a) bundling of D with order π (b) corresponding real labeling of $\mathcal{S}(D)$

Figure 10: Bundling of a storyline drawing: curves added in $\mathcal{S}(D)$ are dashed and a bundling of the crossings is indicated in gray. Note the degenerate bundled crossing at crossing 2.

wiring diagram [15].) Note that the subdivision consists of $|I(D)| + 1$ *columns*, each with $m + 1$ cells, and all cells are either triangular or quadrangular: there are triangles to the left and right of each intersection, and all other cells are quadrangles. The cells in a column are numbered from top to bottom, starting at 1. These numbers are their *row* numbers.

We use the set $\mathcal{L} = \{\times, S, \circ, \downarrow, \uparrow, \updownarrow\}$ of labels. In a fixed bundling of D, each cell satisfies exactly one of the following conditions.

- $\times$ This cell is inside a bundled crossing. (This can only happen if the cell is part of a quadrangular face.)

For cells not inside a bundled crossing, there are five options.

- S This cell is directly left of the π-earliest crossing of a bundled crossing: it "starts" a bundled crossing.
- $\circ$ This cell does not touch the boundary of a bundled crossing, except possibly in a point.
- $\downarrow$ Only the lower boundary of this cell bounds a bundled crossing.
- $\uparrow$ Only the upper boundary of this cell bounds a bundled crossing.
- $\updownarrow$ Both the upper and lower boundary of this cell bound a bundled crossing

We call a function from the set $\mathcal{C}$ of cells to the set $\mathcal{L}$ of labels a *labeling*. A labeling is called *real* if there exists a bundling of D where each cell satisfies the condition of its label. We now observe four necessary properties of real labelings. Afterward, we prove that they are sufficient.

Crossing Property Any crossing must be part of a bundled crossing (though possibly a degenerate one). Consider the six cells surrounding a crossing in $\mathcal{S}(D)$. If the crossing lies in the interior of a bundled crossing, all six cells are labeled $\times$ in the real labeling. Otherwise it lies on the boundary of a bundled crossing. Enumeration reveals the finite set of ways to label these cells that can possibly be real: see Fig. 11. Any other way to label the six cells around the crossing directly contradicts the conditions for the labels.

Column Property Note that in a real labeling, a column of $\mathcal{S}(D)$ can have at most one cell labeled 'S', since by construction only one cell per column is left of a crossing. Now consider the sequence of labels encountered top to bottom in a column, for example $[\circ, \circ, \downarrow, \times, \uparrow, \circ]$ in the third column of Fig. 10b. For any real labeling, this sequence describes being inside ($\times$) and outside ($\circ$) of bundled crossings, with $\downarrow$, $\uparrow$ and $\updownarrow$ marking the transitions, and possibly the label 'S' in place of $\circ$ in one particular cell (left of the crossing). Such sequences without 'S' are walks through the directed graph in Fig. 12; the Column property says that precisely these sequences are allowed, with the addition of also allowing 'S' in the appropriate cell (and only there).

Row Property In any real labeling, horizontally adjacent cells have the same label unless they share a crossing on their boundary (and in that case the change is governed by the Crossing property), where $\circ$ and 'S' are considered the same: the labels $\circ$ and 'S' both describe a cell that does not touch the boundary of a bundled crossing. Such pairs of horizontally adjacent cells must have the same label, because they have the same incidences to any bundled crossings: these incidences can only change at crossings.

Consider for example the sequence of labels encountered left to right in the third row for Fig. 10b: $[\circ, \downarrow, \downarrow, \downarrow, \downarrow, \circ]$. For the Row property, the third and fourth labels must be the same; the other pairs are exempt due to sharing a crossing. The 'S' in the second row is allowed to the right of $\circ$ since the two labels are considered equal for the Row property.

Quadrangle Property In any real labeling, all faces inside a bundled crossing are quadrangles. Therefore, only cells contained in a quadrangular face of D can have the label $\times$.

These properties are necessary for real labelings (as argued above) and, once we fix the leftmost and rightmost columns of the labeling, they are also sufficient.

Lemma 7 *A labeling $\mathcal{C} \to \mathcal{L}$ is real if and only if:*

1. *the first column contains* 'S' *left of its crossing and* $\circ$ *everywhere else,*
2. *the last column contains* $\circ$ *everywhere, and*
3. *the Crossing, Column, Row, and Quadrangle properties hold everywhere.*

The number of bundled crossings in the corresponding bundling equals the number of cells with label 'S'*.*

Proof: First observe that in any real labeling, the first column consists of all $\circ$ except the label 'S' in the unique cell adjacent to a crossing: none of the cells

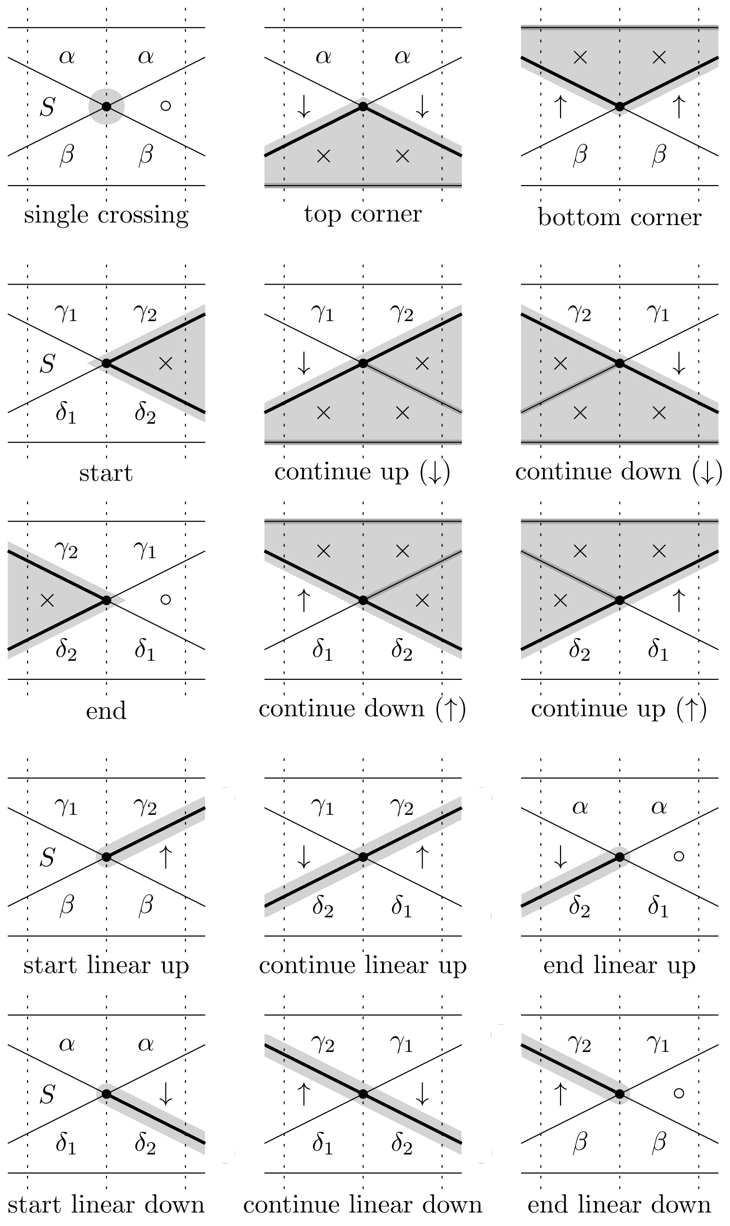


Figure 11: All possible configurations of labels around a crossing on the boundary of a bundled crossing, where the Greek variables may be substituted as follows: $\alpha \in \{\circ, \uparrow\}$, $\beta \in \{\circ, \downarrow\}$, $\gamma_1\gamma_2 \in \{\circ\downarrow, \uparrow\updownarrow\}$, $\delta_1\delta_2 \in \{\circ\uparrow, \downarrow\updownarrow\}$. Multiple occurrences of the same variable within one configuration must be substituted consistently, so for example the two α in the top left configuration must both be $\circ$ or both be $\uparrow$. (If the crossing is in the interior of a boundled crossing, all six cells must be $\times$.)

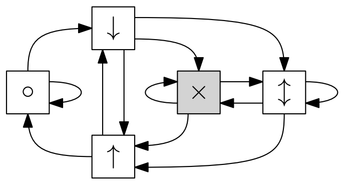


Figure 12: Directed graph for the Column property, reading a column top to bottom; starting nodes are ∘ and ↓. (Further legal columns can be obtained by replacing one occurrence of ∘ by 'S'.) Only the gray state corresponds to cells inside of a bundled crossing.

bound (or are in) a bundled crossing except that the 'S' cell necessarily touches one in a point. Similarly in any real labeling the last column cannot have cells that are contained in a bundled crossing or adjacent to a crossing on their right, so they are all labeled ∘. As argued above, the Crossing, Column, Row, and Quadrangle properties hold everywhere in any real labelling. This establishes one direction of the lemma.

We now show that if a labelling satisfies the three conditions of the lemma, then it is real. Consider a connected component of cells labeled ×. Based on the Row and Crossing properties, the surrounding cells are correctly labeled with arrows and an 'S' label. Call such a connected component a *blob*. We show that each blob is indeed a bundled crossing by Definition 1. (Note that blobs are nondegenerate bundled crossings; we handle degenerate bundled crossings later.)

Let B be a blob. Call a crossing on the boundary of B a *convex corner* if no curve in this crossing goes into the interior of the blob; a *side* if one curve goes into the interior, or; a *reflex corner* if two curves go into the interior. Notice that the "start", "top corner", "bottom corner", and "end" configurations of the Crossing property represent convex corners, and the other configurations with × represent sides. Therefore, a blob cannot have reflex corners. Thus, a blob is a topological disk. Moreover, the first column of $\mathcal{S}(D)$ contains no × labels, so a blob has a unique "start" configuration.

Now we trace the boundary of B, starting at its π-earliest crossing X, and we will find the four frame arcs. Start from X and follow the boundary along one of the curves and call this frame arc e_1. We switch to the next frame arc whenever we encounter a convex corner. This process is repeated until we get back to X, which must happen because the blob is a topological disk and bounded on the right by the last column (which does not contain any ×). Let $e_1, e_2, \ldots, e_k$ be the frame arcs encountered. We only switched from a frame arc to the next at convex corners, and according to the Quadrangle property all faces in the blob are quadrangles. Then $k = 4$, since that is the only way to close a loop around quadrangles using only convex corners. In fact, since all faces in the blob are quadrangles, the crossings inside B form a grid and therefore B is a bundled crossing.

Consider a cell labeled 'S' that does not precede a blob. If the "single crossing" configuration of the Crossing property applies, this correctly describes a singleton bundled crossing. Otherwise, a "start linear up" or "start linear down" configuration must apply (Crossing property). Because of the "continue linear" configurations and the Row property, this must propagate and can only end in an "end linear up" or "end linear down" configuration. This correctly describes a linear bundled crossing.

Therefore, a labeling that starts correctly in the first column, where the properties hold everywhere, and that arrives correctly in the final column, is real. □

In preparation for the runtime bound of Theorem 8, we now bound the number of ways to label a column that are consistent with the Column property.

Lemma 8 *The number of length-n strings over $\mathcal{L}$ that are consistent with the Column property is $O(\varphi^{2n}) \subset O(2.62^n)$, where φ is the golden ratio. The strings can be enumerated with linear-time overhead.*

Proof: Enumerating the walks in the graph from Fig. 12, with the additional option of having 'S' in place of $\circ$ in one particular cell, can be achieved in depth-first fashion using a stack. The optional 'S' at most doubles the number of accepted strings, so we ignore it for the asymptotic analysis and consider only the walks in the graph. We also ignore that there are two starting nodes ($\circ$ and $\downarrow$), since again this only involves a factor two.

Now consider the adjacency matrix A of the graph; here the nodes are given in the order $\circ, \downarrow, \uparrow, \times, \updownarrow$.

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

The characteristic polynomial $\det(A-\lambda I)$ of the matrix A is $-\lambda^5+3\lambda^4-\lambda^3$. Its roots are 0 of multiplicity three, $\frac{1}{2}(3+\sqrt{5}) = \varphi^2$, and φ^{-2}, where φ is the golden ratio. Since the root with the largest absolute value is $\varphi^2 > 1$ and it has multiplicity one, the number of walks of length n is $O(\varphi^{2n})$. The lemma follows. (See, for example, Ardila's treatment [3] of algebraic methods for counting walks.) □

Proof of Theorem 8: We use dynamic programming, moving from left to right by column of $\mathcal{S}(D)$: with $L \in \mathcal{L}^{m+1}$ and i a column, let $f(L, i)$ be the minimum number of 'S' labels in any labeling of the columns up to column i, ending with the labels L for column i. By Lemma 8, there are only $O(\varphi^{2m})$ values of L that satisfy the Column property and they can be enumerated with linear overhead. Each individual $f(L, i)$ can be computed with a constant number of lookups of $f(\,\cdot\,, i-1)$: by the Row property only the three rows adjacent to the crossing between the columns can change and the Crossing property gives a finite set of options for how they can change.

Algorithm 1: Dynamic program for computing the bundled crossing number of storyline visualizations.

```
Input:  Drawing $D$.
Output: Bundled crossing number $\mathrm{bc}^{\mathrm{s}}(D)$.
$C \leftarrow$ Columns of $\mathcal{S}(D)$ numbered 0 to $c$
$L_{\mathrm{start}} \leftarrow$ Entire column is $\circ$, except 'S' left of the crossing of $C[0]$
$F(L_{\mathrm{start}}, 0) \leftarrow 1$                // Dynamic programming data structure
for $i \leftarrow 1$ to $c$ do                                      // $O(c)$ times
    foreach $L \in$ Enum-Single-Columns($C[i]$) do      // $O(\varphi^{2m})$ times
        foreach $L' \in$ Enum-Valid-Predecessors($C[i-1], C[i], L$) do
            $F(L,i) \leftarrow \min\{F(L,i), F(L', i-1)\}$            //↻ $O(1)$ times
        if $L$ contains 'S' then $F(L,i) \leftarrow F(L,i) + 1$
$L_{\mathrm{end}} \leftarrow$ Entire column is $\circ$
return $F(L_{\mathrm{end}}, c)$
```

```
Subroutine: Enum-Single-Columns($c$)
Input:      A single column $c$ of $\mathcal{S}(D)$.
Output:     Enumerates all ways to label the column according to
            the Column property.
```

```
Subroutine: Enum-Valid-Predecessors($c_{\mathrm{pred}}, c_{\mathrm{curr}}, L$)
Input:      Adjacent columns $c_{\mathrm{pred}}$ and $c_{\mathrm{curr}}$ of $\mathcal{S}(D)$, labeling $L$
            for $c_{\mathrm{curr}}$.
Output:     Enumerates all ways to label $c_{\mathrm{pred}}$ according to the four
            properties, given that $c_{\mathrm{curr}}$ is labeled $L$.
```

See Algorithm 1 for pseudocode, where $F(L,i)$ is used to store and look up values of $f(L,i)$; we use the convention that accessing $F(L,i)$ returns ∞ if that value has not been stored yet. The values can be accessed in $O(m)$ time by storing them in a prefix tree (indexed by L) per column. This leads to a total runtime of $O(\varphi^{2m}mc)$. □

If desired, the bundling itself can be read from F. In that case, the algorithm uses $O(\varphi^{2m}mc)$ space: the c prefix trees each store $O(\varphi^{2m})$ items and have height $m+1$. If only the bundled crossing *number* is required, space usage can be improved to $O(\varphi^{2m}m)$ by storing only two columns at a time.

6 Open Problems

Given our new FPT algorithm for simple circular layouts, it would be interesting to improve its runtime and to investigate whether a similar result can be obtained for general simple layouts. A starting point could be the FPT algorithm of Kawarabayashi et al. [23] for computing the usual crossing number of a graph. We also conjecture that it is NP-hard to compute $\mathrm{bc}^{\circ}(G,\pi)$, given a graph G and a vertex order π. It seems plausible to reduce from SortingByTranspositions, but it is difficult to keep the resulting drawings simple.

We remind the reader of the open problem posed by Alam et al. [2] and Fink et al. [16] concerning the computational complexity of $\mathrm{bc}^{\circ}(G)$.

Acknowledgements. We thank Bruno Courcelle for clarifying discussions on the tools available when working with his meta-theorem and in particular MSO_2. We thank the reviewers for their helpful comments, in particular for pointing out a gap in the correctness proof of our algorithm for bundling storylines, which we fixed subsequently.

References

[1] E. Ackerman and R. Pinchasi. On the degenerate crossing number. *Discrete Comput. Geom.*, 49(3):695–702, 2013. doi:10.1007/s00454-013-9493-1.

[2] M. Alam, M. Fink, and S. Pupyrev. The bundled crossing number. In Y. Hu and M. Nöllenburg, editors, *GD*, volume 9801 of *LNCS*, pages 399–412. Springer, 2016. URL: http://arxiv.org/abs/1608.08161, doi:10.1007/978-3-319-50106-2_31.

[3] F. Ardila. Algebraic and geometric methods in enumerative combinatorics. In M. Bóna, editor, *Handbook of Enumerative Combinatorics*, chapter 1.4. CRC Press LLC, Boca Raton, FL, USA, 2015.

[4] A. Arroyo, J. Bensmail, and R. B. Richter. Extending drawings of graphs to arrangements of pseudolines. In S. Cabello and D. Z. Chen, editors, *SoCG*, 2020. To appear. URL: https://arxiv.org/abs/1804.09317.

[5] M. J. Bannister and D. Eppstein. Crossing minimization for 1-page and 2-page drawings of graphs with bounded treewidth. *J. Graph Algorithms Appl.*, 22(4):577–606, 2018. doi:10.7155/jgaa.00479.

[6] U. Bertelè and F. Brioschi. *Nonserial dynamic programming*, volume 91 of *Mathematics in Science & Engineering*. Academic Press, New York, 1972.

[7] H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996. doi:10.1137/S0097539793251219.

[8] S. Chaplick, M. Kryven, G. Liotta, A. Löffler, and A. Wolff. Beyond outerplanarity. In F. Frati and K.-L. Ma, editors, *GD*, volume 10692 of *LNCS*, pages 546–559. Springer, 2018. doi:10.1007/978-3-319-73915-1_42.

[9] S. Chaplick, T. C. van Dijk, M. Kryven, J.-W. Park, A. Ravsky, and A. Wolff. Bundled crossings revisited. In D. Archambault and C. D. Tóth, editors, *GD*, volume 11904 of *LNCS*, pages 63–77. Springer, 2019. doi:10.1007/978-3-030-35802-0_5.

[10] B. Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Inform. Comput.*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.

[11] B. Courcelle and J. Engelfriet. *Graph Structure and Monadic Second-Order Logic: A Language-Theoretic Approach.* Cambridge Univ. Press, 2012. doi:10.1017/CBO9780511977619.

[12] W. Cui, H. Zhou, H. Qu, P. C. Wong, and X. Li. Geometry-based edge clustering for graph visualization. *IEEE Trans. Vis. Comput. Graph.*, 14(6):1277–1284, 2008. doi:10.1109/TVCG.2008.135.

[13] É. C. de Verdière. Computational topology of graphs on surfaces. In C. D. Tóth, J. O'Rourke, and J. E. Goodman, editors, *Handbook of Discrete and Computational Geometry*, chapter 23. CRC Press LLC, Boca Raton, FL, USA, 3rd edition, 2017. URL: `https://www.csun.edu/~ctoth/Handbook/chap23.pdf`.

[14] S. Felsner. On the number of arrangements of pseudolines. *Discrete Comput. Geom.*, 18:257–267, 1997. `doi:10.1007/PL00009318`.

[15] S. Felsner and J. Goodman. Pseudoline arrangements. In J. Goodman, J. O'Rourke, and C. D. Tóth, editors, *Handbook of Discrete and Computational Geometry*, chapter 5. CRC Press LLC, Boca Raton, FL, USA, 3rd edition, 2018.

[16] M. Fink, J. Hershberger, S. Suri, and K. Verbeek. Bundled crossings in embedded graphs. In E. Kranakis, G. Navarro, and E. Chávez, editors, *LATIN*, volume 9644 of *LNCS*, pages 454–468. Springer, 2016. `doi:10.1007/978-3-662-49529-2_34`.

[17] M. Fink, S. Pupyrev, and A. Wolff. Ordering metro lines by block crossings. *J. Graph Algorithms Appl.*, 19(1):111–153, 2015. `doi:10.7155/jgaa.00351`.

[18] E. R. Gansner, Y. Hu, S. North, and C. Scheidegger. Multilevel agglomerative edge bundling for visualizing large graphs. In G. D. Battista, J.-D. Fekete, and H. Qu, editors, *PACIFICVIS*, pages 187–194. IEEE, 2011. `doi:10.1109/PACIFICVIS.2011.5742389`.

[19] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Trans. Vis. Comput. Graphics*, 12(5):741–748, 2006. `doi:10.1109/TVCG.2006.147`.

[20] C. Hurter, O. Ersoy, S. I. Fabrikant, T. R. Klein, and A. C. Telea. Bundled visualization of dynamicgraph and trail data. *IEEE Trans. Vis. Comput. Graphics*, 20(8):1141–1157, 2014. `doi:10.1109/TVCG.2013.246`.

[21] C. Hurter, O. Ersoy, and A. Telea. Graph bundling by kernel density estimation. *Comput. Graph. Forum*, 31:865–874, 2012. `doi:10.1111/j.1467-8659.2012.03079.x`.

[22] K. Kawarabayashi, B. Mohar, and B. A. Reed. A simpler linear time algorithm for embedding graphs into an arbitrary surface and the genus of graphs of bounded tree-width. In *FOCS*, pages 771–780. IEEE, 2008. `doi:10.1109/FOCS.2008.53`.

[23] K. Kawarabayashi and B. Reed. Computing crossing number in linear time. In *STOC*, pages 382–390. ACM, 2007. `doi:10.1145/1250790.1250848`.

[24] F. Lazarus, M. Pocchiola, G. Vegter, and A. Verroust. Computing a canonical polygonal schema of an orientable triangulated surface. In *SoCG*, pages 80–89. ACM, 2001. doi:10.1145/378583.378630.

[25] D. Marx and M. Pilipczuk. Optimal parameterized algorithms for planar facility location problems using Voronoi diagrams. In N. Bansal and I. Finocchi, editors, *ESA*, volume 9294 of *LNCS*, pages 865–877. Springer, 2015. URL: https://arxiv.org/abs/1504.05476, doi:10.1007/978-3-662-48350-3_72.

[26] S. L. Mitchell. Linear algorithms to recognize outerplanar and maximal outerplanar graphs. *Inform. Process. Lett.*, 9(5):229–232, 1979. doi:10.1016/0020-0190(79)90075-9.

[27] B. Mohar. A linear time algorithm for embedding graphs in an arbitrary surface. *SIAM J. Discrete Math.*, 12(1):6–26, 1999. doi:10.1137/S089548019529248X.

[28] B. Mohar. The genus crossing number. *ARS Mathematica Contemporanea*, 2(2):157–162, 2009. doi:10.26493/1855-3974.21.157.

[29] J. Pach and G. Tóth. Degenerate crossing numbers. *Discrete Comput. Geom.*, 41(3):376, 2009. doi:10.1007/s00454-009-9141-y.

[30] S. Pupyrev, L. Nachmanson, S. Bereg, and A. E. Holroyd. Edge routing with ordered bundles. *Comput. Geom. Theory Appl.*, 52:18–33, 2016. doi:10.1016/j.comgeo.2015.10.005.

[31] N. Robertson and P. D. Seymour. Graph minors. III. Planar tree-width. *J. Combin. Theory Ser. B*, 36(1):49–64, 1984. doi:10.1016/0095-8956(84)90013-3.

[32] M. Schaefer. The graph crossing number and its variants: A survey. *Electr. J. Combin.*, Dynamic Survey DS21, 2017. URL: http://www.combinatorics.org/ojs/index.php/eljc/article/view/DS21.

[33] M. Schaefer and D. Štefankovič. The degenerate crossing number and higher-genus embeddings. In E. Di Giacomo and A. Lubiw, editors, *GD*, volume 9411 of *LNCS*, pages 63–74. Springer, 2015. doi:10.1007/978-3-319-27261-0_6.

[34] C. Thomassen. The graph genus problem is NP-complete. *J. Algorithms*, 10(4):568–576, 1989. doi:10.1016/0196-6774(89)90006-0.

[35] T. C. van Dijk, M. Fink, N. Fischer, F. Lipp, P. Markfelder, A. Ravsky, S. Suri, and A. Wolff. Block crossings in storyline visualizations. *J. Graph Algorithms Appl.*, 21(5):873–913, 2017. doi:10.7155/jgaa.00443.

[36] T. C. van Dijk, F. Lipp, P. Markfelder, and A. Wolff. Computing storylines with few block crossings. In F. Frati and K.-L. Ma, editors, *GD*, volume 10692 of *LNCS*, pages 365–378. Springer, 2018. URL: `https://arxiv.org/abs/1709.01055`, `doi:10.1007/978-3-319-73915-1_29`.