






# Bundled Crossings Revisited

Steven Chaplick<sup>1</sup>, Thomas C. van Dijk<sup>1</sup>, Myroslav Kryven<sup>1</sup>,  
Ji-won Park<sup>2</sup>, Alexander Ravsky<sup>3</sup>, and Alexander Wolff<sup>1</sup>

<sup>1</sup> Universität Würzburg, Würzburg, Germany  
steven.chaplick@uni-wuerzburg.de

<sup>2</sup> KAIST, Daejeon, Republic of Korea

<sup>3</sup> Pidstryhach Institute for Applied Problems of Mechanics and Mathematics,  
National Academy of Sciences of Ukraine, Lviv, Ukraine

**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  $k$ ) when we require a simple circular layout. These results make use of the connection between bundled crossings and graph genus.

## 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 [16] 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 [9, 15, 17, 18, 24]. However, bundled drawings have only recently attracted study from a theoretical point of view [1, 11, 13, 14].

Crossing minimization is a fundamental problem in graph drawing [25]. 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 [25, page 35].

*Related Work.* Fink et al. [14] considered bundled crossings (which they called block crossings) in the context of drawing metro maps. A metro network is a

---

The full version of this article is available at ArXiv [5]. M.K. was supported by DAAD; S.C. was supported by DFG grant WO 758/11-1.

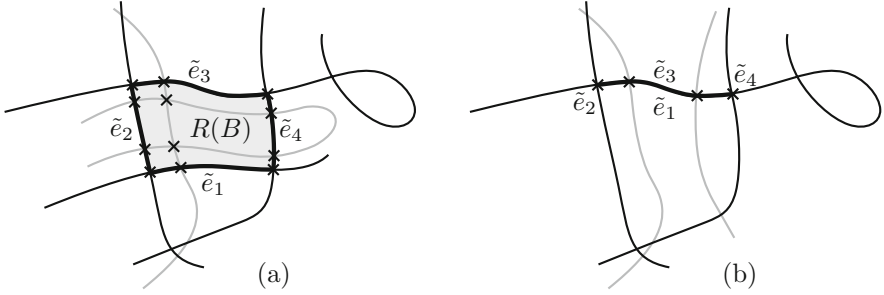
planar graph where vertices are stations and metro lines are simple paths in this graph. These paths representing metro lines can share edges. They enter an edge at one endpoint in some linear order, follow the edge as x-monotone curves (considering the edge as horizontal), and then leave the edge at 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. [11] 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) and can be approximated in a restricted case. Our research builds on recent works of Fink et al. [13] and Alam et al. [1], who extended the notion of bundled crossings from sets of x-monotone curves to general drawings of graphs – details below.

*Notation and Definitions.* In graph drawing, it is common to define a drawing of a graph as a function that maps vertices to points in the plane and edges to Jordan arcs that connect the corresponding points. In this paper, we are less restrictive in that we sometimes allow edges to self-intersect. 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 Jordan region  $R(B)$  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  $D$  (a piece of an edge  $e$  is  $D[e] \cap R(B)$ ); when the edge pieces are not distinct, we define  $R(B)$  not as a Jordan region but as an arc or a point.
- 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  $\tilde{E}_1$  (respectively  $\tilde{E}_2$ ) have a common point in  $R(B)$ .

Our definition is similar to that of Alam et al. [1] but defines the Jordan region  $R(B)$  more precisely. We call the sets  $\tilde{E}_1$  and  $\tilde{E}_2$  of edge pieces *bundles*



**Fig. 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

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. 1(b). 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  $bc(G)$  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  $bc'(G)$ . 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*  $bc^\circ(G)$  and  $bc^{o\prime}(G)$  of a graph  $G$ .

Fink et al. [13] showed that it is NP-hard to compute the minimum number of bundled crossings that a given drawing of a graph 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 of bundled crossings in the case of a *fixed 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. [1] showed that  $bc'(G)$  equals the orientable genus of  $G$ , which in general is NP-hard to compute [26]. They also showed that there is a graph  $G$  with  $bc'(G) \neq bc(G)$  by proving that  $bc'(K_6) = 1 < bc(K_6)$ . As it turns out, the two problem variants differ in the circular setting, too (see Fig. 2 and Observation 2). For computing  $bc(G)$  and  $bc^\circ(G)$ , Alam et al. [1] gave an algorithm whose approximation factor depends on the density of the graph. They posed the existence of an FPT algorithm for  $bc^\circ(G)$  as an open question.

*Our Contribution.* As some graphs  $G$  have  $bc'(G) \neq bc(G)$  (see Fig. 2), Fink et al. [13] posed the complexity of computing the bundled crossing number  $bc(G)$  of a given graph  $G$  as an open problem. We settle this in Sect. 2 as follows:

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

Our main result, which we prove in Sect. 3, resolves an open question of Alam et al. [1] 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  $\text{bc}^\circ(G) \leq k$ , 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 [3] 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 (see the full version [5]). 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 ( $\text{MSO}_2$ ), Courcelle’s Theorem [7] (stated as Theorem 5 in Sect. 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  $\text{MSO}_2$  formula is somewhat more difficult in our case due to the more complex way our regions interact with our special set of edges.

## 2 Computing $\text{bc}(G)$ Is NP-Hard

For a given graph  $G$ , finding a drawing with the fewest bundled crossings resembles computing the *orientable genus*<sup>1</sup>  $g(G)$  of  $G$ . In fact, Alam et al. [1] showed that  $\text{bc}'(G) = g(G)$ . Thus, deciding  $\text{bc}'(G) = k$  for some  $k$  is NP-hard and that it is FPT in  $k$ , since the same holds for deciding  $g(G) = k$  [19, 23, 26].

**Theorem 3** [1]. *For every graph  $G$  with genus  $k$ , it holds that  $\text{bc}'(G) = k$ .*

To show this, Alam et al. [1] first showed that a drawing with  $k$  bundled crossings can be lifted onto a surface of genus  $k$ , and thus  $\text{bc}'(G) \geq g(G)$ :

**Observation 1** [1]. *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. 7.*

<sup>1</sup> I.e., computing the fewest *handles* to attach to the sphere so that  $G$  can be drawn on the resulting surface without any crossings.

Then, to see that  $\text{bc}'(G) \leq \text{g}(G)$ , Alam et al. [1] used the *fundamental polygon* representation (or *polygonal schema*) [10] of a drawing on a genus- $g$  surface. More precisely, the sides of the polygon are numbered in circular order  $a_1, b_1, a'_1, b'_1, \dots, a_g, b_g, a'_g, b'_g$ ; 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 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. [1] 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-loops and pairs of edges crossing multiple times, e.g., Alam et al. [1, Lemma 1] showed that  $\text{bc}(K_6) = 2$  while  $\text{bc}'(K_6) = \text{g}(K_6) = 1$ .

We show that the problem 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  $\text{g}(G)$  of  $G$  by showing that  $\text{bc}(G') = \text{g}(G)$ , with Observation 1 in mind.

Consider the embedding of  $G$  onto the genus- $\text{g}(G)$  surface. By a result of Lazarus et al. [21, Theorem 1], we can construct a fundamental polygon representation of the embedding so that its boundary intersects with edges of the graph  $O(\text{g}(G)|E(G)|)$  times. Note that each edge piece outside the polygon intersects each other edge piece at most once; 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 subdividing of 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  $\text{g}(G) \leq |E(G)|$ , by subdividing edges further whenever necessary, we obtain a drawing of  $G'$ . Our subdivisions keep the integrity of all bundled crossings, so  $\text{bc}(G') \leq \text{g}(G)$ . On the other hand, since subdividing edges does not affect the genus,  $\text{g}(G) = \text{g}(G') = \text{bc}'(G') \leq \text{bc}(G')$ .  $\square$

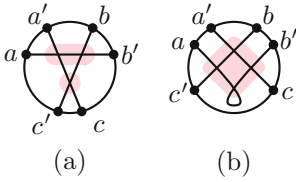
### 3 FPT Algorithms for Computing $\text{bc}^{o'}(G)$ and $\text{bc}^o(G)$

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

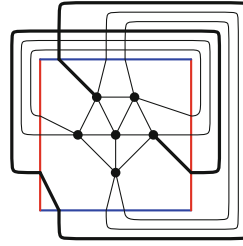
**Observation 2.**  $\text{bc}^{o'}(K_{3,3}) = 1$  but  $\text{bc}^o(K_{3,3}) > 1$ .

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

Similarly to computing  $\text{bc}'(G)$ , we use compute  $\text{bc}^{o'}(G)$  via computing genus.



**Fig. 2.**  $bc^\circ(K_{3,3}) \neq bc^{o'}(K_{3,3})$ ; see Observation 2



**Fig. 3.**  $K_6$  drawn in a fundamental square; the self-intersecting edge is bold [1, Fig. 2].

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

*Proof (Sketch).* This follows from the fact that  $bc^{o'}(G) = g(G^*)$  where  $G^*$  is a graph with a vertex  $v^*$  adjacent to every vertex of  $G$  (see the full version [5] and the  $2^{g^{O(1)}} n$  time algorithm for genus [19]). □

To prove our main result (Theorem 2) we develop an algorithm that tests whether  $bc^\circ(G) = k$  in FPT time with respect to  $k$ . Our algorithm is inspired by recent works on circular layouts with at most  $k$  crossings [3] and circular layouts where each edge is crossed at most  $k$  times [6]. 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. (For basic definitions of both treewidth and  $MSO_2$  logic, see the appendix of the full version.)

**Theorem 5 (Courcelle [7,8]).** *For any integer  $t \geq 0$  and any  $MSO_2$  formula  $\psi$  of length  $\ell$ , 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  $\ell$ .*

We proceed along the lines of Bannister and Eppstein [3], 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.

### 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. 7. 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  $e_1, \dots, e_4$  respectively (here  $(e_1, e_3)$  and  $(e_2, e_4)$  frame the two bundles and  $e_i = u_i v_i$ ). 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. 4) to preserve the topology of the surface. Thus,  $D_{\mathcal{S}}$  is crossing-free.

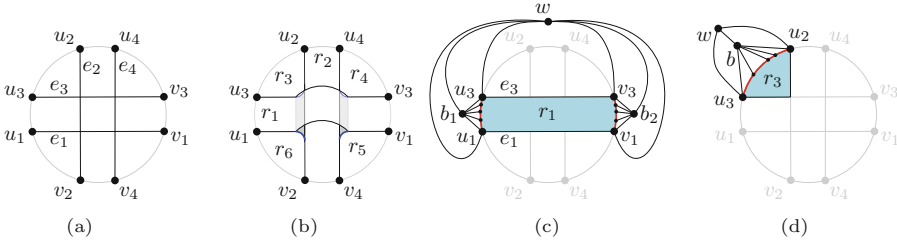
We now cut  $\mathcal{S}$  into *components* (maximal connected subsets) along the frame edges and corner-cuts of each bundled crossing, resulting in a subdivision  $\Omega$  of  $\mathcal{S}$ .

We use  $D_{\Omega}$  to denote the sub-drawing of  $D_{\mathcal{S}}$  on  $\Omega$ , i.e.,  $D_{\Omega}$  is missing the frame edges since these have been cut out. We now consider the components of  $\Omega$ . Notice that every edge of  $D_{\Omega}$  is contained in one component of  $\Omega$ . In order for a component  $s$  of  $\Omega$  to contain an edge  $e$  of  $D_{\Omega}$ ,  $s$  must have both endpoints of  $e$  on its boundary. With this in mind we focus on the components of  $\Omega$  where each one has a vertex of  $G$  on its boundary and call such components *regions*. Observe that a crossing in  $D$  which does not involve a frame edge corresponds, in  $D_{\Omega}$ , 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, and so a set  $F$  of four frame edges. By  $V(F)$  we denote the set of vertices incident to frame edges. Via the construction above, we obtain the subdivided surface  $\Omega$ ; see Fig. 4. 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  $r_3, \dots, r_6$  be the regions each bounded by one edge from one pair and one from the other pair; see Fig. 4. These are all the regions of  $\Omega$ . 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  $G \setminus V(F)$  including the edges connecting it to vertices of  $V(F)$  is drawn in  $D_{\Omega}$  in some region of  $\Omega$ . In this sense, for each region  $r$  of  $\Omega$ , we use  $G_r$  to denote the subgraph of  $G$  induced by the components of  $G \setminus V(F)$  contained in  $r$ , including 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:  $\{r_1, r_2\}$  and  $\{r_3, r_4, r_5, r_6\}$ . Consider a region of the first type, say  $r_1$ ; see Fig. 4. Observe that  $r_1$  is a topological disk, i.e.,  $G_{r_1}$  is outerplanar. Moreover,  $G_{r_1}$  has a special outerplanar drawing where on the boundary of  $r_1$  (in clockwise order) we see the frame edge  $e_1$ ,



**Fig. 4.** (a) Bundled crossing; (b) regions, corner-cuts in blue; (c),(d) augmented graphs  $G_{r_1}^*$  and  $G_{r_3}^*$  consist of the edges of  $G_{r_1}$  and  $G_{r_3}$  (blue), augmentation vertices and edges (black) (Color figure online)

the vertices mapped to the  $(u_1, u_3)$ -arc, the frame edge  $e_3$ , then the vertices mapped to the  $(v_3, v_1)$ -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<sup>2</sup>. 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. 4. 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  $C = (v_1, b_2, v_3, u_3, b_1, u_1, v_1)$ , edges from  $b_1$  to the vertices on the  $(u_1, u_3)$ -arc, and edges from  $b_2$  to the vertices on the  $(v_3, v_1)$ -arc (both including the end points). 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  $(h, b_1, u_3, h)$ . But then it can easily be moved into  $C$ . Similarly, a component that is incident only to  $u_3$  and  $v_3$  can end up in the face  $(h, u_3, v_3, h)$ , 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  $(u_3, u_2)$ -arc of the disk subtended by the two frame edges  $e_3$  and  $e_2$  bounding the region  $r_3$ . We augment similarly as for  $r_1$ ; see Fig. 4. For the augmented graph  $G_{r_3}^*$ , we add to  $G_{r_3}$  a boundary vertex  $b$  neighboring all vertices on the  $(u_3, u_2)$ -arc and a hub vertex  $h$  adjacent to  $u_2$ ,  $b$ , and  $u_3$ . Again,  $G_{r_3}^*$  is planar since  $G_{r_3}$  is outerplanar due to  $r_3$  being a topological disk. 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  $u_3 u_2 h$ , the edge  $u_3 u_2$  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  $r_3$ .

<sup>2</sup> This augmentation may sound overly complicated, but is written as to easily generalize to more bundled crossings.



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 [22], 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  $\text{bc}^\circ(G) = 1$ , it must have treewidth at most 10 (and this can be checked in linear time using an algorithm of Bodlaender [4]).

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  $e_1, e_2, \dots, e_\beta$  (this set is denoted  $F$ ) and  $v_1, \dots, v_\xi$  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  $r_1, \dots, r_\gamma$  of the subdivided surface  $\Omega$  so that the following properties hold (note that the frame vertices partition the boundary of the disk underlying  $\Omega$  into  $\eta \leq 8$  (possibly degenerate) arcs  $p_1, \dots, p_\eta$  where each such  $p_j$  is contained in a unique region  $r_{i_j}$  of  $\Omega$ ):

1.  $E(G)$  is partitioned into  $E_0, E_1, \dots, E_\gamma$ , where  $E_0 = \{f_1, \dots, f_\beta\}$ .
2.  $V(G)$  is partitioned into  $V_0, V_1, \dots, V_\eta$ , where  $V_0 = \{u_1, \dots, u_\xi\}$ .
3. The mapping  $u_i \leftrightarrow v_i$  and  $f_i \leftrightarrow e_i$  defines an isomorphism between the subgraph of  $G$  formed by  $(V_0, E_0)$  and graph  $(V_F, F)$ .
4. No vertex in  $V(G) \setminus V_0$  has incident edges  $e \in E_i, e' \in E_j$  for  $i \neq j$ .
5. For each  $v \in V_0$ , and each edge  $e$  incident to  $v$ , exactly one of the following is true: (i)  $e \in E_0$  or (ii)  $e \in E_i$  and  $v$  is on the boundary of  $r_i$ .
6. For each  $v \in V_j$ , all edges incident to  $v$  belong to  $E_{i_j}$ .
7. For each region  $r_i$ , let  $G_i$  be the graph  $(V_0 \cup \bigcup_{j: i_j=i} V_j, E_i)$  (i.e., the subgraph that is to be drawn in  $r_i$ ), and let  $G_i^*$  be the corresponding augmented graph (i.e., as in  $\star$  above). Each  $G_i^*$  is planar.

We now describe the algorithm to test for a simple circular drawing with one bundled crossing. First we check that 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  $\text{MSO}_2$  (see the full version [5]). The only property that is not obviously expressible is the planarity of  $G_i^*$ . 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  $\text{MSO}$  formula  $\text{MINOR}_H$  so that for all graphs  $G$ , it holds that  $G \models \text{MINOR}_H$  if and only if  $G$  contains  $H$  as a minor [8, Corollary 1.14]. Additionally, each  $G_i^*$  can be expressed as an *MSO-transduction*<sup>3</sup> of  $G$  and our variables (our transduction can be thought of as a kind of 2-copying transduction). Thus, by [8, Theorem 7.10] using the transduction and the  $\text{MSO}$  formula testing planarity, we can construct an  $\text{MSO}_2$  formula  $\iota$  so that when  $G \models \iota$ ,  $G_i^*$  is planar for every  $i$ . Therefore, Properties 1–7 can be expressed as an  $\text{MSO}_2$  formula  $\psi$  and, by

<sup>3</sup> For the formalities of transductions, see the book of Courcelle and Engelfriet [8, Section 1.7.1, and Definitions 7.6 and 7.25].



**Fig. 5.** Configurations for  $p = 6$ : (a)  $D_A(p)$ , (b)  $D_B(p)$ , and (c) induced by a hole

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 algorithm.

### 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  $\Omega$ , and the corresponding set of regions. The key ingredient above was that every region was a topological disk. 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  $D_A(p)$  and  $D_B(p)$  of a matching with  $p + 1$  edges  $f_0, f_1, \dots, f_p$  (see, e.g., Fig. 5) such that

- edge  $f_i$  crosses only  $f_{i-1 \bmod p+1}$  and  $f_{i+1 \bmod p+1}$  for  $i = 0, \dots, p$ ;
- the endpoints of each edge  $f_i$ ,  $i = 1, \dots, p - 2$ , are inside the cycle  $C$  formed by the crossing points and the edge-pieces between these crossing points;
- both endpoints of  $f_{p-1}$ , only one endpoint of  $f_0$ , and only one endpoint of  $f_p$  are contained in  $C$  in the drawing  $D_A(p)$ ;
- only one endpoint of  $f_{p-1}$ , only one endpoint of  $f_0$ , and no endpoints of  $f_p$  are contained in  $C$  in the drawing  $D_B(p)$ .

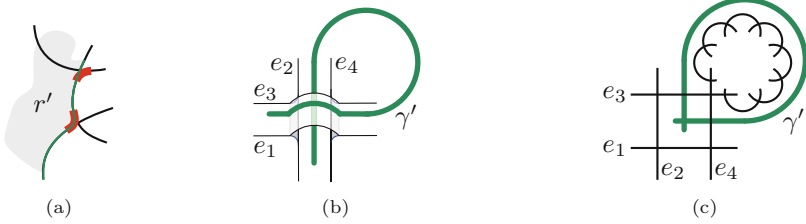
Note that the partial drawings  $D_A(p)$  and  $D_B(p)$  differ only in how the last edge is drawn with respect to the previous edge. Arroyo et al. [2, Theorem 1.2] showed that such partial drawings are obstructions for pseudolinearity, that is, they cannot be part of any pseudoline arrangement. Therefore, neither of these partial drawings can be *completed* to a simple circular drawing, that is, the 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 1.** *For a matching with  $p + 1$  edges  $f_0, f_1, \dots, f_p$ , neither the partial drawing  $D_A(p)$  nor  $D_B(p)$  can be completed to a simple circular drawing.*

Using this lemma we can now prove the following statement.

**Lemma 2.** *Each region  $r$  of  $\Omega$  is a topological disk<sup>4</sup>.*

<sup>4</sup> We slightly abuse this notion to also mean a simply connected set.



**Fig. 6.** (a) Projection  $r'$  of the region  $r$  and its boundary (green, the corner-cuts are in blue) onto the disk of the drawing  $D$  (b) projection  $\gamma'$  of a Jordan arc  $\gamma$  that goes over and under the same handle; (c) profile of edges of the projected boundary of  $r$  enclosed by the loop made by  $\gamma'$  form a partial drawing  $D_A(p)$ . (Color figure online)

*Proof.* First, we show that no region of  $\Omega$  includes part of both a handle and its *undertunnel*, that is, the part of the surface over which the handle was built. Then we will show that a region also does not include holes.

Let  $r$  be a region of the surface subdivision  $\Omega$ . The boundary of this region is formed by pieces of frame edges that were lifted on the surface  $\mathcal{S}$  as described above and the additional corner-cuts as illustrated in Fig. 4 in red. Consider the projection  $r'$  of  $r$  and its boundary on the drawing  $D$  in the plane. Note that the projected boundary either follows an edge in  $D$  or switches to some another edge via a corner-cut at an intersection point; see Fig. 6(a).

Suppose now, for a contradiction, that  $r$  contains both a handle and its undertunnel corresponding to the same bundled crossing  $B = ((e_1, e_3), (e_2, e_4))$ .

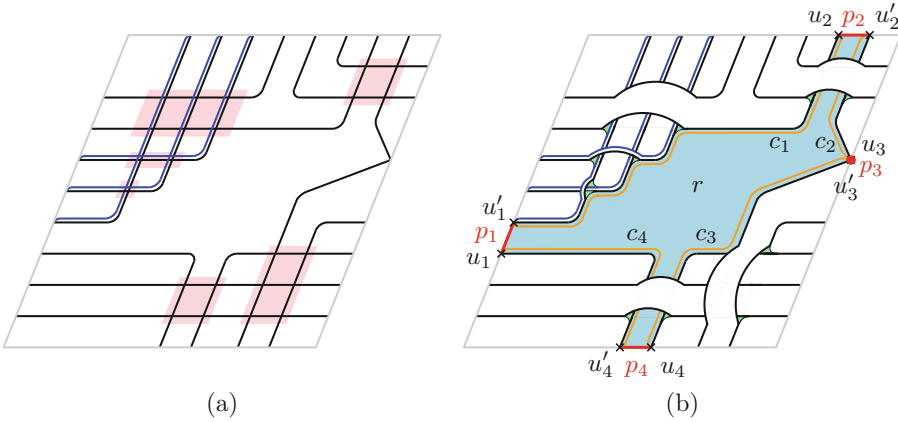
Then there is a Jordan arc  $\gamma \subset r$  going over and under this handle making a loop; see Fig. 6(b). Note that the orthogonal projection  $\gamma'$  of  $\gamma$  on the disk of the drawing  $D$  self-intersects. The profile of edges along the projected boundary of  $r$  that is enclosed by  $\gamma'$  then inevitably contains a partial drawing  $D_A(p)$ ; see Fig. 6(c). And according to Lemma 1, 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 the boundary edges around this hole would give a partial drawing of edges as illustrated in Fig. 5(c). Therefore, the region  $r$  is a proper topological disk.  $\square$

The next lemma concerning treewidth is a direct consequence of Lemma 2.

**Lemma 3.** *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 2, the removal of their endpoints breaks up the graph into outerplanar components. The treewidth of an outerplanar graph is at most two [22]. 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  $8k + 2$ .  $\square$



**Fig. 7.** (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). (Color figure online)

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

*Proof (of Theorem 2).* We use Lemma 2 and extend the algorithm of Sect. 3.2.

Suppose  $G$  has a circular drawing  $D$  with at most  $k$  bundled crossings. In  $D$  we see the set  $F$  of (up to)  $4k$  frame edges of these bundled crossings. As before,  $F$  together with  $D$  defines a subdivided topological surface  $\Omega$  containing a set of regions  $R$ . As in the one bundled crossing case, each edge of  $G$  is 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 will refer to Fig. 7 for an example. By Lemma 2, each region  $r$  is a topological disk and as such its graph  $G_r$  is outerplanar with a quite special drawing  $D_r$  described as follows. In particular, if we trace the boundary of  $r$  in clockwise order, we see that it is made up of arcs  $p_1, \dots, p_\alpha$  of  $\mathcal{S}$ , marked in red in Fig. 7(b) (such arcs can degenerate to single points), and Jordan arcs  $c_1, \dots, c_\alpha$ , traced in orange in Fig. 7(b), each of which connects two such arcs of the disk. For  $i \in \{1, \dots, \alpha\}$ , let  $u_i$  and  $u'_i$  be the end points of  $p_i$ , in clockwise order. So  $u'_i$  and  $u_{i+1}$  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  $i \in \{1, \dots, \alpha\}$ , if  $u'_i$  and  $u_{i+1}$  (where  $u_{\alpha+1}$  is  $u_1$ ) are not adjacent, we add an edge between them. If  $p_i$  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, for technical reasons (see the full version), we identify  $b_i$  with  $u_i$  and  $u'_i$ .

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 Sect. 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  $b_1, \dots, b_\alpha$  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  $\beta$ -edge  $k$ -bundled crossing drawing  $D_F$  (of the graph formed by  $F$ ), whose corresponding surface  $\mathcal{S}$  consists of regions  $r_1, \dots, r_\gamma$  (note:  $\gamma \leq 2\beta \leq 8k$ ) so that Properties 1–7 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  $k$ ) [4]. It then enumerates all possible simple drawings of at most  $4k$  edges in the circle<sup>5</sup>. For each drawing, it further enumerates the possible ways to form  $k$  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<sub>2</sub> formula  $\varphi$  (see the full version) to express Properties 1–7. Finally, since  $G$  has treewidth at most  $8k + 2$ , we can apply Courcelle's theorem on  $(G, \varphi)$ .  $\square$

## 4 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. [20] for computing the usual crossing number of a graph.

**Acknowledgements.** We thank Bruno Courcelle for clarifying discussions on the tools available when working with his meta-theorem and in particular MSO<sub>2</sub>.

## References

1. Alam, M.J., Fink, M., Pupyrev, S.: The bundled crossing number. In: Hu, Y., Nöllenburg, M. (eds.) GD 2016. LNCS, vol. 9801, pp. 399–412. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-50106-2\\_31](https://doi.org/10.1007/978-3-319-50106-2_31). <http://arxiv.org/abs/1608.08161>
2. Arroyo, A., Bensmail, J., Richter, R.B.: Extending drawings of graphs to arrangements of pseudolines. ArXiv report (2018). <https://arxiv.org/abs/1804.09317>

<sup>5</sup> 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 [12].

3. Bannister, M.J., Eppstein, D.: Crossing minimization for 1-page and 2-page drawings of graphs with bounded treewidth. *J. Graph Algorithms Appl.* **22**(4), 577–606 (2018). <https://doi.org/10.7155/jgaa.00479>
4. Bodlaender, H.L.: A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.* **25**(6), 1305–1317 (1996). <https://doi.org/10.1137/S0097539793251219>
5. Chaplick, S., van Dijk, T.C., Kryven, M., won Park, J., Ravsky, A., Wolff, A.: Bundled crossings revisited. ArXiv report (2019). <https://arxiv.org/abs/1812.04263>
6. Chaplick, S., Kryven, M., Liotta, G., Löffler, A., Wolff, A.: Beyond outerplanarity. In: Frati, F., Ma, K.-L. (eds.) *GD 2017*. LNCS, vol. 10692, pp. 546–559. Springer, Cham (2018). [https://doi.org/10.1007/978-3-319-73915-1\\_42](https://doi.org/10.1007/978-3-319-73915-1_42)
7. Courcelle, B.: The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Inform. Comput.* **85**(1), 12–75 (1990). [https://doi.org/10.1016/0890-5401\(90\)90043-H](https://doi.org/10.1016/0890-5401(90)90043-H)
8. Courcelle, B., Engelfriet, J.: *Graph Structure and Monadic Second-Order Logic: A Language-Theoretic Approach*. Cambridge Univ Press, Cambridge (2012)
9. Cui, W., Zhou, H., Qu, H., Wong, P.C., Li, X.: Geometry-based edge clustering for graph visualization. *IEEE Trans. Vis. Comput. Graph.* **14**(6), 1277–1284 (2008). <https://doi.org/10.1109/TVCG.2008.135>
10. de Verdière, É.C.: Computational topology of graphs on surfaces. In: Tóth, C.D., O’Rourke, J., Goodman, J.E. (eds.) *Handbook of Discrete and Computational Geometry*, 3rd edn., chap. 23. CRC Press LLC, Boca Raton (2017)
11. van Dijk, T.C., Fink, M., Fischer, N., Lipp, F., Markfelder, P., Ravsky, A., Suri, S., Wolff, A.: Block crossings in storyline visualizations. *J. Graph Algorithms Appl.* **21**(5), 873–913 (2017). <https://doi.org/10.7155/jgaa.00443>
12. Felsner, S.: On the number of arrangements of pseudolines. In: *SoCG*, pp. 30–37. ACM (1996). <https://doi.org/10.1145/237218.237232>
13. Fink, M., Hershberger, J., Suri, S., Verbeek, K.: Bundled crossings in embedded graphs. In: Kranakis, E., Navarro, G., Chávez, E. (eds.) *LATIN 2016*. LNCS, vol. 9644, pp. 454–468. Springer, Heidelberg (2016). [https://doi.org/10.1007/978-3-662-49529-2\\_34](https://doi.org/10.1007/978-3-662-49529-2_34)
14. Fink, M., Pupyrev, S., Wolff, A.: Ordering metro lines by block crossings. *J. Graph Algorithms Appl.* **19**(1), 111–153 (2015). <https://doi.org/10.7155/jgaa.00351>
15. Gansner, E.R., Hu, Y., North, S., Scheidegger, C.: Multilevel agglomerative edge bundling for visualizing large graphs. In: Battista, G.D., Fekete, J.D., Qu, H. (eds.) *PACIFICVIS*, pp. 187–194. IEEE (2011). <https://doi.org/10.1109/PACIFICVIS.2011.5742389>
16. Holten, D.: Hierarchical edge bundles: visualization of adjacency relations in hierarchical data. *IEEE Trans. Vis. Comput. Graph.* **12**(5), 741–748 (2006). <https://doi.org/10.1109/TVCG.2006.147>
17. Hurter, C., Ersoy, O., Fabrikant, S.I., Klein, T.R., Telea, A.C.: Bundled visualization of dynamicgraph and trail data. *IEEE Trans. Vis. Comput. Graph.* **20**(8), 1141–1157 (2014). <https://doi.org/10.1109/TVCG.2013.246>
18. Hurter, C., Ersoy, O., Telea, A.: Graph bundling by kernel density estimation. *Comput. Graph. Forum* **31**, 865–874 (2012). <https://doi.org/10.1111/j.1467-8659.2012.03079.x>
19. Kawarabayashi, K., Mohar, B., Reed, B.A.: A simpler linear time algorithm for embedding graphs into an arbitrary surface and the genus of graphs of bounded tree-width. In: *FOCS*, pp. 771–780. IEEE (2008). <https://doi.org/10.1109/FOCS.2008.53>

20. Kawarabayashi, K., Reed, B.: Computing crossing number in linear time. In: STOC, pp. 382–390. ACM (2007). <https://doi.org/10.1145/1250790.1250848>
21. Lazarus, F., Pocchiola, M., Vegter, G., Verroust, A.: Computing a canonical polygonal schema of an orientable triangulated surface. In: SoCG, pp. 80–89. ACM (2001). <https://doi.org/10.1145/378583.378630>
22. Mitchell, S.L.: Linear algorithms to recognize outerplanar and maximal outerplanar graphs. *Inform. Process. Lett.* **9**(5), 229–232 (1979). [https://doi.org/10.1016/0020-0190\(79\)90075-9](https://doi.org/10.1016/0020-0190(79)90075-9)
23. Mohar, B.: A linear time algorithm for embedding graphs in an arbitrary surface. *SIAM J. Disc. Math.* **12**(1), 6–26 (1999)
24. Pupyrev, S., Nachmanson, L., Bereg, S., Holroyd, A.E.: Edge routing with ordered bundles. *Comput. Geom. Theory Appl.* **52**, 18–33 (2016). <https://doi.org/10.1016/j.comgeo.2015.10.005>
25. Schaefer, M.: The graph crossing number and its variants: a survey. *Electr. J. Combin. Dynamic Survey DS21* (2017). <http://www.combinatorics.org/ojs/index.php/eljc/article/view/DS21>
26. Thomassen, C.: The graph genus problem is NP-complete. *J. Algorithms* **10**(4), 568–576 (1989). [https://doi.org/10.1016/0196-6774\(89\)90006-0](https://doi.org/10.1016/0196-6774(89)90006-0)