



# Recognizing 2-Layer and Outer $k$ -Planar Graphs

Yasuaki Kobayashi 

Hokkaido University, Sapporo, Japan

Yuto Okada 

Nagoya University, Japan

Alexander Wolff 

Universität Würzburg, Germany

---

## Abstract

The *crossing number* of a graph is the least number of crossings over all drawings of the graph in the plane. Computing the crossing number of a given graph is NP-hard, but fixed-parameter tractable (FPT) with respect to the natural parameter. Two well-known variants of the problem are *2-layer crossing minimization* and *circular crossing minimization*, where every vertex must lie on one of two *layers*, namely two parallel lines, or a circle, respectively. In both cases, edges are drawn as straight-line segments. Both variants are NP-hard, but admit FPT-algorithms with respect to the natural parameter.

In recent years, in the context of beyond-planar graphs, a local version of the crossing number has also received considerable attention. A graph is *k-planar* if it admits a drawing with at most  $k$  crossings per edge. In contrast to the crossing number, recognizing *k-planar* graphs is NP-hard even if  $k = 1$  and hence not likely to be FPT with respect to the natural parameter  $k$ .

In this paper, we consider the two above variants in the local setting. The *k-planar* graphs that admit a straight-line drawing with vertices on two layers or on a circle are called *2-layer k-planar* and *outer k-planar* graphs, respectively. We study the parameterized complexity of the two recognition problems with respect to the natural parameter  $k$ . For  $k = 0$ , the two classes of graphs are exactly the caterpillars and outerplanar graphs, respectively, which can be recognized in linear time. Two groups of researchers independently showed that outer 1-planar graphs can also be recognized in linear time [Hong et al., Algorithmica 2015; Auer et al., Algorithmica 2016]. One group asked explicitly whether outer 2-planar graphs can be recognized in polynomial time.

Our main contribution consists of XP-algorithms for recognizing 2-layer *k-planar* graphs and outer *k-planar* graphs, which implies that both recognition problems can be solved in polynomial time for every fixed  $k$ . We complement these results by showing that recognizing 2-layer *k-planar* graphs is XNLP-complete and that recognizing outer *k-planar* graphs is XNLP-hard. This implies that both problems are  $W[t]$ -hard for every  $t$  and that it is unlikely that they admit FPT-algorithms. On the other hand, we present an FPT-algorithm for recognizing 2-layer *k-planar* graphs where the order of the vertices on one layer is specified.

**2012 ACM Subject Classification** Mathematics of computing → Graph theory

**Keywords and phrases** 2-layer *k-planar* graphs, outer *k-planar* graphs, recognition algorithms, local crossing number, bandwidth, FPT, XNLP, XP,  $W[t]$

**Digital Object Identifier** 10.4230/LIPIcs.SoCG.2025.65

**Related Version** Full Version: <https://arxiv.org/abs/2412.04042>

**Funding** *Yasuaki Kobayashi*: Supported by JSPS KAKENHI Grant Numbers JP23K28034, JP24H00686, and JP24H00697.

*Yuto Okada*: Supported by JST SPRING, Grant Number JPMJSP2125 and JSPS KAKENHI, Grant Number JP22H00513 (Hirotaka Ono).

**Acknowledgements** We thank Hirotaka Ono and the AFSA project (Creation and Organization of Innovative Algorithmic Foundations for Social Advancement) for supporting our work. We thank Boris Klemz and Marie Diana Sieper for useful discussions.



© Yasuaki Kobayashi, Yuto Okada, and Alexander Wolff;  
licensed under Creative Commons License CC-BY 4.0

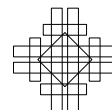
41st International Symposium on Computational Geometry (SoCG 2025).

Editors: Oswin Aichholzer and Haitao Wang; Article No. 65; pp. 65:1–65:16

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



## 1 Introduction

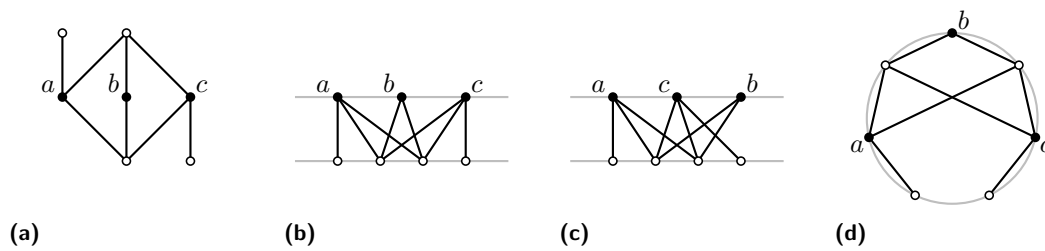
When evaluating the quality of a graph drawing, one of the established metrics is the number of crossings, whose importance is supported by user experiments [42]. Unfortunately, computing the crossing number of a given graph, that is, the minimum number of crossings over all drawings of the graph, is NP-hard [27], even for graphs that become planar after removal of a single edge [14]. On the other hand, the problem is fixed-parameter tractable (FPT) with respect to the natural parameter, that is, the number of crossings [29, 33]. Many variants of the crossing number have been studied; see Schaefer’s survey [44]. Two variants with geometric restrictions have attracted considerable attention: *2-layer crossing minimization* and *circular* (or *convex*, or *1-page*) *crossing minimization*, where the placement of the vertices is restricted to two parallel lines (called *layers*) and to a circle, respectively. In both cases, edges are drawn as straight-line segments. Circular crossing minimization is NP-hard, but admits FPT-algorithms with respect to the natural parameter [6, 34]. In practice, often the so-called *sifting heuristic* is used [7]. Circular crossing minimization can be seen as a special case of a *book embedding problem*, where vertices must lie on a straight line, the *spine* of the book, and each edge must be drawn on one of a given number of halfplanes called *pages* whose intersection is the spine. In this setting, crossing minimization is interesting even if the order of the vertices along the spine is given [8, 38].

The 2-layer variant comes in two settings: *one-sided crossing minimization* (OSCM) and *two-sided crossing minimization* (TSCM). In OSCM, the input consists of a (bipartite) graph and a linear order for the vertices on one side of the bipartition; the task is to find a linear order for the vertices on the other side that minimizes the total number of crossings. In TSCM, the linear orders on both layers can be chosen freely. OSCM is an important step in the so-called Sugiyama framework for drawing hierarchical graphs [46], that is, graphs where each vertex is assigned to a specific layer. OSCM was the topic of the Parameterized Algorithms and Computational Experiments Challenge (PACE<sup>1</sup>) 2024. Both OSCM and TSCM are NP-hard; OSCM even for the disjoint union of 4-stars [39] and for trees [19]. On the positive side, OSCM admits a subexponential FPT-algorithm; it runs in  $O(k2^{\sqrt{2k}} + n)$  time [35]. TSCM also admits an FPT-algorithm; it runs in  $2^{O(k)} + n^{O(1)}$  time [36].

In the context of beyond-planar graphs, a local version of the crossing number has also received considerable attention [18, 22]. A graph is *k-planar* if it admits a drawing with at most  $k$  crossings per edge. The *local crossing number* of a graph is the smallest  $k$  such that the graph is  $k$ -planar. The recognition of 1-planar graphs has long been known to be NP-hard [28]. Later, it turned out that the recognition of  $k$ -planar graphs is NP-hard for every  $k$  [47]. Hence, it is unlikely that FPT- or XP-algorithms exist with respect to the natural parameter  $k$ . On the other hand, recognizing 1-planar graphs is fixed-parameter tractable with respect to tree-depth and cyclomatic number [5]. The problem remains NP-hard, however, for graphs of bounded bandwidth (and hence, pathwidth and treewidth). The local crossing number has also been studied in the context of book embeddings [1, 37].

In this paper, we study the above-mentioned geometric restrictions, but with respect to the *local* crossing number. The resulting graph classes are called *2-layer k-planar* graphs and *outer k-planar* graphs; see Figure 1. The former were studied by Angelini, Da Lozzo, Förster, and Schneck [3]. Among others, they gave bounds on the edge density of these graphs and characterized 2-layer  $k$ -planar graphs with the maximum edge density for  $k \in \{2, 4\}$ . They concluded that “the general recognition and characterization of 2-layer  $k$ -planar graphs

<sup>1</sup> <https://pacechallenge.org/2024/>



■ **Figure 1** Drawings of the same bipartite graph with optimal local crossing number in different settings: (a) planar drawing, (b) 2-layer 2-planar drawing without restriction, (c) 2-layer 3-planar drawing where the vertex order on the upper layer is fixed, (d) outer 1-planar drawing.

remain important open problems”. According to Schaefer’s survey [44] on crossing numbers, Kainen [32] introduced the “local outerplanar crossing number”, which minimizes, over all circular drawings, the largest number of crossings along any edge. Outer  $k$ -planar graphs have been studied by Pach and Tóth [40], who showed that any outer  $k$ -planar graph with  $n$  vertices has at most  $4.1\sqrt{kn}$  edges. For  $k \leq 3$ , they established a better bound  $(k+3)(n-2)$ , which is tight for  $k \in \{1, 2\}$ . For  $k \geq 5$ , the constant factor was later improved to  $\sqrt{243/40} \approx 2.46$  [2].

We study the parameterized complexity of the two recognition problems with respect to the natural parameter  $k$ . For  $k = 0$ , the two classes of graphs are exactly the caterpillars and outerplanar graphs, respectively, which can be recognized in linear time. There are also linear-time algorithms for recognizing outer 1-planar graphs [4, 30]. The authors of [30] posed the existence of polynomial-time algorithms for recognizing outer 2-planar graphs as an open problem. A partial answer has been given by Hong and Nagamochi [31], showing that *full* outer 2-planar graphs can be recognized in linear time. Outer  $k$ -planar drawings are *full* if no crossing appears on the boundary of the outer face. The authors of [16] generalized this result and showed that, for every integer  $k$ , full outer  $k$ -planarity is testable in  $O(f(k) \cdot n)$  time, for a computable function  $f$ . They also showed that outer  $k$ -planar graphs can be recognized in quasi-polynomial time, which implies that, for every integer  $k$ , testing outer  $k$ -planarity is not NP-hard unless the Exponential-Time Hypothesis fails.

**Parameterized complexity.** We assume that the reader is familiar with basic concepts in parameterized complexity theory (see [17, 20, 25] for definitions of these concepts). Zehavi [48] gives a survey specifically on the parameterized analysis of crossing minimization problems. The class XNLP consists of all parameterized problems that can be solved non-deterministically in time  $f(k)n^{O(1)}$  and space  $f(k) \log n$ , where  $f$  is some computable function,  $n$  is the input size, and  $k$  is the parameter. A parameterized problem  $L_2 \subseteq \Sigma^* \times \mathbb{N}$  is said to be XNLP-hard if for any  $L_1 \in \text{XNLP}$ , there is a *parameterized logspace reduction* from  $L_1$  to  $L_2$ , that is, there is an algorithm  $\mathcal{A}$  and computable functions  $f$  and  $g$  that satisfy the following: Given  $(x_1, k_1) \in \Sigma^* \times \mathbb{N}$ , the algorithm  $\mathcal{A}$  computes  $(x_2, k_2) \in \Sigma^* \times \mathbb{N}$  such that  $(x_1, k_1) \in L_1$  if and only if  $(x_2, k_2) \in L_2$ ,  $k_2 \leq g(k_1)$ , and  $\mathcal{A}$  runs in space  $O(f(k_1) + \log |x_1|)$ . A parameterized problem is said to be XNLP-complete if it is XNLP-hard and belongs to XNLP. The class XNLP contains the class  $W[t]$  for every  $t \geq 1$  [13]. Moreover, Pilipczuk and Wrochna [41] conjectured that an XNLP-hard problem does not admit an algorithm that runs in  $n^{f(k)}$  time and  $f(k) \cdot n^{O(1)}$  space for a computable function  $f$ , where  $n$  is the size of the instance and  $k$  is the parameter. We refer to [13, 23] for more information.

Recently, the authors of [10] showed the first graph-drawing problem to be XNLP-complete, namely *ordered level planarity*, parameterized by the number of levels. Ordered level planarity is a restricted version of level planarity, where for each level, the vertices on that level are given in order (and the problem is to route the edges in a y-monotone and crossing-free way).

**Our contribution.** We present XP-algorithms for recognizing 2-layer  $k$ -planar graphs and outer  $k$ -planar graphs, which implies that both recognition problems can be solved in polynomial time for every fixed  $k$ ; see Sections 4.1 and 5.1, respectively. This solves the open problem regarding the recognition of outer 2-planar graphs posed by the authors of [30]. We complement these results by showing that recognizing 2-layer  $k$ -planar graphs is XNLP-complete even for trees (Section 4.2) and that recognizing outer  $k$ -planar graphs is XNLP-hard (Section 5.3). This implies that both problems are  $W[t]$ -hard for every  $t$  [13] and that it is unlikely that they admit FPT-algorithms. On the other hand, we present an FPT-algorithm for recognizing 2-layer  $k$ -planar graphs where the order of the vertices on one layer is specified; see Figure 1c and Section 3.1. We prove that two *edge-weighted* versions of this problem are NP-hard; see Section 3.2. Finally, we show that the local circular crossing number cannot be approximated even for graphs that are almost trees (that is, graphs with feedback vertex number 1); see Section 5.2. We conclude with open problems; see Section 6.

The proofs of statements marked with a (clickable)  $\star$  are in the full version.

## 2 Preliminaries

Let  $G$  be a graph. We let  $V(G)$  and  $E(G)$  denote the sets of vertices and edges of  $G$ , respectively. For a vertex  $v$  of  $G$ , let  $N_G(v)$  be the set of neighbors of  $v$  in  $G$ , and let  $\delta_G(v)$  be the set of edges incident to  $v$ . For  $U \subseteq V(G)$ , we define  $N_G(U) = (\bigcup_{v \in U} N_G(v)) \setminus U$ ,  $N_G[U] = N_G(U) \cup U$ , and  $\delta_G(U) = \bigcup_{v \in U} \delta_G(v)$ . We may omit the subscript  $G$  when it is clear from the context. The subgraph of  $G$  induced by  $U \subseteq V(G)$  is denoted by  $G[U]$ . A vertex is called a *leaf* if it has exactly one neighbor.

We use  $[\ell]$  as shorthand for  $\{1, 2, \dots, \ell\}$ . Let  $n = |V(G)|$ , and let  $\sigma: V(G) \rightarrow [n]$  be a linear order of the vertices of  $G$  (i.e., a bijection between  $V(G)$  and  $[n]$ ). For  $\{u, v\} \in E$ , the *stretch* of edge  $\{u, v\}$  in  $\sigma$  is defined as  $|\sigma(u) - \sigma(v)|$ . The *bandwidth* of  $\sigma$  (with respect to  $G$ ) is the maximum stretch of an edge of  $G$  in  $\sigma$ . The *bandwidth* of  $G$ , denoted by  $\text{bw}(G)$ , is the minimum integer  $k$  such that  $G$  has a linear order  $\sigma$  of  $V$  with bandwidth at most  $k$ .

We define a *circular drawing* of a graph  $G$  to be a cyclic order  $D = (v_1, \dots, v_n)$  of  $V(G)$ . We say that an edge  $\{v_i, v_j\}$  with  $i < j$  *pierces* a pair of (not necessarily adjacent) vertices  $\{v_{i'}, v_{j'}\}$  with  $i' < j'$  if either  $1 \leq i < i' < j < j' \leq n$  or  $1 \leq i' < i < j' < j \leq n$  holds. In particular, if  $\{v_{i'}, v_{j'}\}$  is an edge of  $G$ , we say that  $\{v_i, v_j\}$  *crosses*  $\{v_{i'}, v_{j'}\}$ . For an edge  $e$ , let  $\text{cr}_D(e)$  denote the number of edges that cross  $e$  in  $D$ . A circular drawing  $D$  is  *$k$ -planar* (or an *outer  $k$ -planar drawing*) if every edge crosses at most  $k$  edges in  $D$ . Since whether two edges cross is determined only by the cyclic vertex order, in this paper, we allow the edges to be drawn arbitrarily.

Let  $G$  be a bipartite graph with  $V(G) = X \cup Y$ ,  $X \cap Y = \emptyset$ , and  $E(G) \subseteq X \times Y$ . A *2-layer drawing* of  $G$  is a pair  $D = (<_X, <_Y)$  of (strict) linear orders  $<_X$  and  $<_Y$  defined on  $X$  and on  $Y$ , respectively. A *crossing* in  $D$  is defined by a pair of edges  $\{x, y\}$  and  $\{x', y'\}$  with distinct endpoints  $x, x' \in X$  and distinct endpoints  $y, y' \in Y$  such that  $x <_X x'$  and  $y' <_Y y$ . The notation  $\text{cr}_D(e)$  is defined as above. Moreover, for two distinct edges  $e$  and  $e'$ , let  $\text{cr}_D(e, e') = 1$  if  $e$  crosses  $e'$ ;  $\text{cr}_D(e, e') = 0$  otherwise. For distinct  $x, x' \in X$ , we say that  $x$  is *to the left of*  $x'$  in  $D$  if  $x <_X x'$ . Equivalently, we say that  $x'$  is *to the right of*  $x$ . The *leftmost* (resp. *rightmost*) of  $X$  in  $D$  is the smallest (resp. largest) vertex in  $X$  under the linear order  $<_X$ . We also use these notions for vertices in  $Y$ . For an integer  $k \geq 0$ , a 2-layer drawing  $D$  of  $G$  is said to be  *$k$ -planar* if, for each edge  $e$  of  $G$ ,  $\text{cr}_D(e) \leq k$ . For (not necessarily disjoint) vertex sets  $A, B \subseteq X \cup Y$  and 2-layer drawings  $D_A$  of  $G[A]$  and  $D_B$  of  $G[B]$ , we say that  $D_A$  is *compatible* with  $D_B$  (or equivalently  $D_B$  is compatible with  $D_A$ ) if, for every pair  $\{z, z'\} \subseteq A \cap B$  of vertices that both are in the same set of the bipartition  $Z \in \{X, Y\}$ , we have that  $z <_Z z'$  in  $D_A$  if and only if  $z <_Z z'$  in  $D_B$ .

### 3 Recognizing 2-Layer $k$ -Planar Graphs – The One-Sided Case

In this section, we design an FPT-algorithm for recognizing 2-layer  $k$ -planar graphs when the order of the vertices on one layer is given as input. The problem is defined as follows.

**Problem:** ONE-SIDED  $k$ -PLANARITY

**Input:** A bipartite graph  $(X \cup Y, E)$ , an integer  $k \geq 0$ , and a linear order  $<_X$  of  $X$ .

**Question:** Does  $Y$  admit a linear order  $<_Y$  such that  $(<_X, <_Y)$  is a 2-layer  $k$ -planar drawing of  $(X \cup Y, E)$ ?

**Degree reduction.** Let  $G = (X \cup Y, E)$  be a bipartite graph, and let  $k$  be a non-negative integer. We describe two simple reduction rules that yield an equivalent instance of ONE-SIDED  $k$ -PLANARITY where every vertex in  $X$  has degree at most  $2k + 2$ .

► **Observation 1** ( $\star$ ). *Let  $G = (X \cup Y, E)$  be a bipartite graph that contains a vertex with more than  $2k + 2$  non-leaf neighbors. Then  $G$  is not 2-layer  $k$ -planar.*

► **Lemma 2** ( $\star$ ). *Let  $(G, <_X, k)$  be an instance of ONE-SIDED  $k$ -PLANARITY. If  $G$  contains a vertex  $v \in X$  with  $\deg(v) > 2k + 2$  and with a leaf neighbor  $y \in Y$ , then  $(G, <_X, k)$  is a YES-instance if and only if  $(G - y, <_X, k)$  is a YES-instance.*

Hence, in the following, we assume that every vertex in  $X$  has degree at most  $2k + 2$ .

#### 3.1 An FPT-Algorithm

Let  $n$  be the number of vertices in  $G$ . In this section, we prove the following result.

► **Theorem 3.** *ONE-SIDED  $k$ -PLANARITY can be solved in time  $2^{O(k \log k)} n^{O(1)}$ , that is, ONE-SIDED  $k$ -PLANARITY is fixed-parameter tractable when parameterized by  $k$ .*

We assume that  $G$  has no isolated vertices; otherwise, we simply remove them. For a 2-layer drawing  $D$  of a subgraph  $G'$  of  $G$ , we say that  $D$  respects  $<_X$  if, for every  $x, x' \in V(G')$ , it holds that  $x$  is to the left of  $x'$  in  $D$  if and only if  $x <_X x'$ .

We first give a simpler algorithm with running time  $2^{O(k^2 \log k)} n^{O(1)}$ . Let  $x_1, \dots, x_{|X|}$  be the vertices of  $X$  appearing in this order in  $<_X$ . If  $|X| < 2k + 1$ , then  $|Y| \leq 2k(2k + 2)$ , and we can simply enumerate all the possible 2-layer drawings in time  $2^{O(k^2 \log k)} n^{O(1)}$ . Thus, we assume  $|X| \geq 2k + 1$ . Let  $\ell = 2k$ . For  $i \in [|X| - \ell]$ , let  $X_{\leq i} = \{x_1, \dots, x_{i+\ell}\}$  and  $X_i = \{x_i, \dots, x_{i+\ell}\}$ . Correspondingly, let  $G_{\leq i} = G[N[X_{\leq i}]]$  and  $G_i = G[N[X_i]]$ . Our algorithm recursively decides whether  $G_{\leq i}$  admits a 2-layer  $k$ -planar drawing that extends a prescribed partial drawing  $D$  of  $G_i$ . To be more precise, let  $D$  be a 2-layer  $k$ -planar drawing of  $G_i$  that respects  $<_X$ . Given  $i \in [|X| - \ell]$ , a partial drawing  $D$  of  $G_i$  respecting  $<_X$ , and a function  $\chi: \delta(X_i) \rightarrow \{0, \dots, k\}$ , we define a Boolean value  $\mathbf{draw}(i, D, \chi)$  to be **true** if and only if  $G_{\leq i}$  admits a 2-layer  $k$ -planar drawing  $D_{\leq i}$  such that

- $D_{\leq i}$  is compatible with  $D$  and
- for every edge  $e \in \delta(X_i)$ , it holds that  $\chi(e) = \text{cr}_{D_{\leq i}}(e)$ .

Our goal is to compute  $\mathbf{draw}(|X| - \ell, D, \chi)$  for some partial drawing  $D$  of  $G_{|X| - \ell}$  and  $\chi$ , which is done by the following dynamic programming algorithm.

For the base case  $i = 1$ , we compute the table entries by brute force: For each possible 2-layer  $k$ -planar drawing  $D$  of  $G_1$  respecting  $<_X$  and for every function  $\chi: \delta(X_1) \rightarrow \{0, \dots, k\}$ , we set  $\mathbf{draw}(1, D, \chi) = \mathbf{true}$  if  $\chi(e) = \text{cr}_D(e)$  for every  $e \in \delta(X_1)$ ; otherwise, we set  $\mathbf{draw}(1, D, \chi) = \mathbf{false}$ . Now we show that, in any 2-layer  $k$ -planar drawing, if two edges have endpoints in  $X$  that are far apart, then the edges do not cross.

► **Lemma 4** (\*). If  $e = (x_p, y)$  and  $f = (x_q, y')$  are distinct edges such that  $p + \ell < q$ , then, in every 2-layer  $k$ -planar drawing  $D = (\prec_X, \prec_Y)$  of  $G$ , it holds that  $y <_Y y'$  or that  $y = y'$ .

This suggests the following recurrence for the dynamic program.

► **Lemma 5.** Let  $2 \leq i \leq |X| - \ell$ , let  $D$  be a 2-layer  $k$ -planar drawing of  $G_i$  respecting  $\prec_X$ , and let  $\chi: \delta(X_i) \rightarrow \{0, \dots, k\}$  such that  $\chi(e) = \text{cr}_D(e)$  for every  $e \in \delta(x_{i+\ell})$ . Then,

$$\text{draw}(i, D, \chi) = \bigvee_{D_{i-1}, \chi_{i-1}} \text{draw}(i-1, D_{i-1}, \chi_{i-1}),$$

where the  $D_{i-1}$  are taken over all 2-layer  $k$ -planar drawings of  $G_{i-1}$  that are compatible with  $D$ , and the  $\chi_{i-1}$  are taken over all functions  $\delta(X_{i-1}) \rightarrow \{0, \dots, k\}$  that satisfy

$$\chi_{i-1}(e) = \chi(e) - \sum_{f \in \delta(x_{i+\ell})} \text{cr}_D(e, f)$$

for every edge  $e \in \delta(X_i) \cap \delta(X_{i-1})$ .

**Proof.** Suppose that  $\text{draw}(i, D, \chi) = \text{true}$ . Then, there is a 2-layer  $k$ -planar drawing  $D_{\leq i}$  of  $G_{\leq i}$  that is compatible with  $D$  and, for every edge  $e \in \delta(X_i)$ , it holds that  $\chi(e) = \text{cr}_{D_{\leq i}}(e)$ . We need to show that there exists a triplet  $t = (i-1, D_{i-1}, \chi_{i-1})$  such that  $\text{draw}(t) = \text{true}$ . Let  $D_{\leq i-1}$  and  $D_{i-1}$  be subdrawings of  $D_{\leq i}$  induced by  $G_{\leq i-1}$  and  $G_{i-1}$ , respectively. Then  $D_{\leq i-1}$  is compatible with  $D_{i-1}$ . By Lemma 4, edges incident to  $x_{i+\ell}$  do not cross edges incident to  $x_{i-1}$ . Thus, for every edge  $e \in \delta(x_{i-1})$ , we have  $\text{cr}_{D_{\leq i}}(e) = \text{cr}_{D_{\leq i-1}}(e)$ . Moreover, every edge  $e \in \delta(X_i) \cap \delta(X_{i-1})$  has exactly  $\chi(e) - \sum_{f \in \delta(x_{i+\ell})} \text{cr}_D(e, f)$  crossings in  $D_{\leq i-1}$ . Define  $\chi_{i-1}$  by setting  $\chi_{i-1}(e) = \text{cr}_{D_{\leq i-1}}(e)$  for every edge  $e \in \delta(X_{i-1})$ . Then, by definition,  $\text{draw}(i-1, D_{i-1}, \chi_{i-1}) = \text{true}$  since  $D_{\leq i-1}$  is a 2-layer  $k$ -planar drawing of  $G_{\leq i-1}$  compatible with  $D_{i-1}$  and, for every  $e \in \delta(X_{i-1})$ , it trivially holds that  $\chi_{i-1}(e) = \text{cr}_{D_{\leq i-1}}(e)$ .

We omit the converse direction, which readily follows by reversing the above argument. ◀

Our algorithm evaluates the recurrence of Lemma 5 in a dynamic programming manner. To see the runtime bound, observe that, for each  $i \in [|X| - \ell]$ , the number of possible 2-layer  $k$ -planar drawings of  $G_i$  is upper bounded by  $|N(X_i)|! \leq ((\ell + 1) \cdot (2k + 2))! = 2^{O(k^2 \log k)}$  and the number of possible functions from  $\delta(X_i)$  to  $\{0, \dots, k\}$  is upper bounded by  $(k + 1)^{|\delta(X_i)|} = 2^{O(k^2 \log k)}$ . Hence, we can evaluate the recurrence in time  $2^{O(k^2 \log k)} n^{O(1)}$ .

We can improve the exponential dependency of our running time as follows. Instead of fixing the “window size” to  $2k + 1$ , for every  $i$ , we dynamically take the smallest  $\ell_i$  such that  $\delta(\{x_i, \dots, x_{i+\ell_i}\})$  consists of at least  $2k + 1$  edges. It is easy to verify that Lemma 4 (and hence Lemma 5) still holds for this dynamic window size. Since the degree of every vertex in  $X$  is at most  $2k + 2$ , we have that  $|\delta(\{x_i, \dots, x_{i+\ell_i}\})| \leq 4k + 2$ . This improves the running time to  $2^{O(k \log k)} n^{O(1)}$ , completing the proof of Theorem 3.

### 3.2 NP-Hardness of the Weighted Version

We can generalize ONE-SIDED  $k$ -PLANARITY to weighted settings. Let  $G = (X \cup Y, E)$  be a bipartite graph, and let  $w: E \rightarrow \mathbb{N}_{>0}$  be an edge-weight function. A 2-layer drawing  $D$  of  $(G, w)$  is said to be  $k$ -planar if, for each edge  $e$  of  $G$ , it holds that

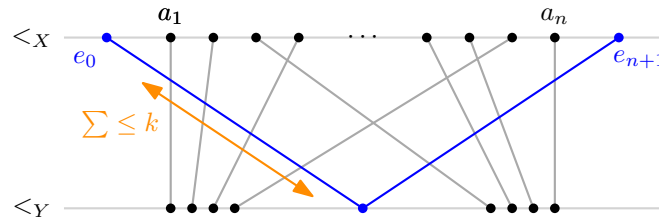
$$\sum_{f \text{ crosses } e \text{ in } D} w(f) \leq k. \quad (1)$$

It is straightforward to extend our algorithm to this weighted setting. Although we believe that ONE-SIDED  $k$ -PLANARITY is NP-hard, we can only show the following weaker result.



► **Theorem 6** (★). *The weighted ONE-SIDED  $k$ -PLANARITY is (weakly) NP-hard under (1).*

**Proof (sketch).** The claim is shown by performing a reduction from PARTITION, which is known to be (weakly) NP-hard [26]. The problem asks, given a set of  $n$  integers  $A = \{a_1, \dots, a_n\}$ , whether the set can be partitioned into two subsets of equal sum. We construct a bipartite graph  $G$  consisting of a path of length 2 with two edges  $e_0$  and  $e_{n+1}$  of weight 1, and  $n$  isolated edges with weight proportional to the integers in  $A$ . By appropriately defining the order  $<_X$  on  $X$ , we can ensure that each isolated edge crosses either  $e_0$  or  $e_{n+1}$ ; see Figure 2. Setting  $k$  properly induces two balanced subsets of  $A$ . ◀



■ **Figure 2** The graph  $G$  and  $<_X$  we construct.  $\sum$  is the sum of weights of edges that cross  $e_0$ .

We remark that there is another reasonable definition of crossings in a weighted graph: A 2-layer drawing is defined to be  $k$ -planar if, for each edge  $e \in E$ , it holds that

$$\sum_{f \text{ crosses } e \text{ in } D} w(e) \cdot w(f) \leq k. \quad (2)$$

By making  $w(e_0)$  and  $w(e_{n+1})$  sufficiently large, a similar reduction will work.

► **Remark 7.** The weighted ONE-SIDED  $k$ -PLANARITY is (weakly) NP-hard under (2).

## 4 Recognizing 2-Layer $k$ -Planar Graphs – The Two-Sided Case

The algorithm in Theorem 3 exploits the prescribed order  $<_X$  on  $X$ , which is not specified in the two-sided case. This difference is reflected by the parameterized complexity of the two problems. The two-sided case turns out to be XNLP-hard, meaning that it is unlikely to be fixed-parameter tractable. On the other hand, we design a polynomial-time algorithm for the two-sided case, provided that  $k$  is fixed. We use this algorithm also to show that the problem is contained in XNLP. Formally, the problem is defined as follows.

**Problem:** TWO-SIDED  $k$ -PLANARITY

**Input:** A bipartite graph  $G = (X \cup Y, E)$  and an integer  $k \geq 0$ .

**Question:** Does  $G$  admit a 2-layer  $k$ -planar drawing?

### 4.1 An XP-Algorithm

To solve TWO-SIDED  $k$ -PLANARITY, we extend the algorithm for ONE-SIDED  $k$ -PLANARITY presented in Section 3. Let  $G = (X \cup Y, E)$  be a bipartite graph, let  $n$  be the number of vertices of  $G$ , and let  $k \in \mathbb{N}$ . We assume that  $G$  is connected; otherwise, the problem can be solved independently for each connected component. Moreover, by applying Observation 1 and applying Lemma 2 first to  $X$  and then to  $Y$ , we assume that every vertex has degree at most  $2k + 2$ . Our algorithm employs a dynamic programming approach analogous to that

presented in Section 3. Instead of a “window”, we specify a subset  $X_i \subseteq X$  of  $\ell + 1 = 2k + 1$  vertices, which plays the same role as the window  $\{x_i, \dots, x_{i+\ell}\}$ . However, this subset does not specify the graph  $G_{\leq i}$  on the left of the window, preventing us from defining the same type of subproblems as there. We overcome this obstacle by applying an idea similar to that of Saxe [43] for recognizing bandwidth- $k$  graphs. To properly define the subproblems, we observe that  $N[X_i]$  separates the subdrawings of the components of  $G[V(G) \setminus N[X_i]]$  into left and right parts.

► **Lemma 8.** *Let  $D = (<_X, <_Y)$  be a 2-layer  $k$ -planar drawing of  $G$ , and let  $S \subseteq X$  be a set of  $\ell + 1$  vertices that appears consecutively in  $D$ . Let  $x$  and  $x'$  be the leftmost vertex and the rightmost vertex of  $S$  in  $D$ , respectively. Then, for each component  $C$  of  $G[V(G) \setminus N[S]]$ , the vertices in  $C \cap X$  are either entirely to the left of  $x$  or entirely to the right of  $x'$ .*

**Proof.** Suppose that  $C$  has two vertices  $u, v \in X \setminus S$  such that  $u$  is to the left of  $x$  and  $v$  is to the right of  $x'$  in  $D$ . Let  $P$  be a path between  $u$  and  $v$  in  $G[C]$ . We can assume that  $P$  has exactly two edges,  $e$  and  $f$ . Observe that each edge incident to a vertex in  $S$  crosses either  $e$  or  $f$ . Since each vertex in  $S$  has at least one incident edge, at least one of  $e$  and  $f$  involves more than  $k$  crossings. ◀

Suppose that  $G$  has a 2-layer  $k$ -planar drawing  $D = (<_X, <_Y)$ . For a family  $\mathcal{D} \subseteq 2^{V(G)}$ , we use  $\mathcal{D}^X$  as shorthand for  $\bigcup_{C \in \mathcal{D}} C \cap X$ . Let  $x_1, \dots, x_{|X|}$  be the vertices of  $X$  appearing in this order in  $<_X$ . For  $1 \leq i \leq |X| - \ell$ , let  $\mathcal{C}_i$  be the set of connected components in  $G[V(G) \setminus N[\{x_i, \dots, x_{i+\ell}\}]]$ . By Lemma 8, we have  $\mathcal{C}^X = \{x_1, \dots, x_{i-1}\}$  for some  $\mathcal{C} \subseteq \mathcal{C}_i$ .

Now, we can formally define our subproblems. Let  $S \subseteq X$  with  $|S| = \ell + 1$ , let  $D$  be a 2-layer  $k$ -planar drawing of  $G[N[S]]$ , let  $\chi: \delta(S) \rightarrow \{0, \dots, k\}$ , and let  $\mathcal{C} \subseteq \mathcal{C}_S$ , where  $\mathcal{C}_S$  is the set of components in  $G[V(G) \setminus N[S]]$ . We define a Boolean value  $\text{draw}(S, D, \chi, \mathcal{C})$  to be **true** if and only if there is a 2-layer  $k$ -planar drawing  $D^*$  of  $G[N[S \cup \mathcal{C}^X]]$  such that

- $D^*$  is compatible with  $D$  and
- for every edge  $e \in \delta(S)$ , it holds that  $\chi(e) = \text{cr}_{D^*}(e)$ .

Hence,  $G$  has a 2-layer  $k$ -planar drawing if and only if  $\text{draw}(S, D, \chi, \mathcal{C}_S) = \text{true}$  for some  $S \subseteq X$ ,  $D$ ,  $\chi$ , and  $\mathcal{C}_S$ .

To compute the values  $\text{draw}(S, D, \chi, \mathcal{C})$  for  $S \subseteq X$  with  $|S| = \ell + 1$ ,  $D$ ,  $\chi$ , and  $\mathcal{C} \subseteq \mathcal{C}_S$ , we first compute the base cases where  $\mathcal{C} = \emptyset$  and then the other cases in ascending order of  $|S \cup \mathcal{C}^X|$ . This can be done by using a recurrence similar to the one in Lemma 5.

To see the running time bound of the above algorithm, observe that the number of possible choices for  $S$ ,  $D$ ,  $\chi$ , and  $\mathcal{C}$  is at most

$$\sum_{S \subseteq X} |S|! \cdot |N(S)|! \cdot (k+1)^{|\delta(S)|} \cdot 2^{|\mathcal{C}_S|} = n^{\ell+1} \cdot 2^{O(k^2 \log k)} \cdot 2^{|\mathcal{C}_S|}.$$

The third factor can be bounded by  $2^{O(k^3)}$  as follows: Since  $G$  is connected, each connected component of  $G[V(G) \setminus N[S]]$  contains at least one vertex in  $N(N(S)) \setminus S$ . This implies that the number of components in  $G[V(G) \setminus N[S]]$  is at most  $|N(N(S)) \setminus S| \leq (2k+2)(2k+1)^2$ .

► **Theorem 9.** *TWO-SIDED  $k$ -PLANARITY can be solved in time  $2^{O(k^3)} n^{2k+O(1)}$ , that is, TWO-SIDED  $k$ -PLANARITY is polynomial-time solvable when  $k$  is fixed.*

The above algorithm easily turns into a non-deterministic algorithm that runs in polynomial time and space  $k^{O(1)} \log n$ , which implies the following.

► **Corollary 10** (\*). *TWO-SIDED  $k$ -PLANARITY is in XNLP.*



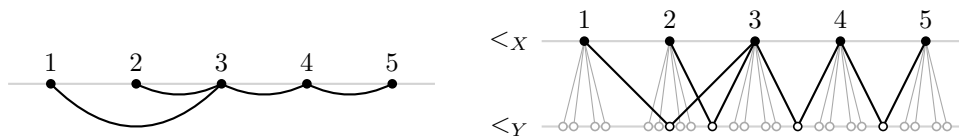
## 4.2 XNLP-Completeness

To complement the positive result in the previous subsection, we show that TWO-SIDED  $k$ -PLANARITY is XNLP-hard even on trees. In contrast to our result, TSCM can be solved in polynomial time on trees [45].

► **Theorem 11** (★). *TWO-SIDED  $k$ -PLANARITY is XNLP-complete w.r.t.  $k$  even on trees.*

**Proof (sketch).** Membership in XNLP follows from Corollary 10. We prove the claim by showing a parameterized logspace reduction from BANDWIDTH, which is known to be XNLP-hard even on trees [11,13]. Let  $T$  be a tree. We subdivide each edge  $e$  of  $T$  once by introducing a vertex  $w_e$ , and we add  $\ell$  leaves adjacent to each original vertex of  $T$  for some  $\ell = \Theta(b^2)$ . Let  $G$  be the graph obtained in this way. Next, we show that  $\text{bw}(T) \leq b$  if and only if  $G$  has a 2-layer  $k$ -planar drawing for some  $k = \Theta(b^3)$ .

Let  $X = V(T)$ , let  $Y = V(G) \setminus X$ , and let  $\sigma$  be a vertex order of  $T$  with bandwidth  $b$ . Define a vertex order  $<_X$  on  $X$  by setting  $<_X = \sigma$ . Since the stretch of each edge in  $\sigma$  is at most  $b$ , there are at most  $b-1$  vertices between its endpoints. This implies that we can place the vertices in  $Y$  so that there will be only  $O(b^3)$  crossings per edge; see Figure 3. Conversely, in any 2-layer  $k$ -planar drawing of  $G$ , the endpoints of every edge  $e = \{u, v\}$  of  $T$  are close to each other, as each vertex between  $u$  and  $v$  causes at least  $\ell$  crossings on the path  $(u, w_e, v)$ . Hence, the order on  $X$  turns into a vertex order of  $T$  with bandwidth at most  $b$ . ◀



■ **Figure 3** A minimum-bandwidth order of  $T$  and the 2-layer drawing of  $G$  that we construct.

## 5 Recognizing Outer $k$ -Planar Graphs

In this section, we discuss the parameterized complexity of recognizing outer  $k$ -planar graphs.

**Problem:** OUTER  $k$ -PLANARITY

**Input:** A graph  $G$  with  $n$  vertices and an integer  $k$ .

**Question:** Does  $G$  admit an outer  $k$ -planar drawing?

### 5.1 An XP-Algorithm

In this subsection, we show our main result, an XP-algorithm for OUTER  $k$ -PLANARITY with respect to  $k$ . Note that a graph is outer  $k$ -planar if and only if its biconnected components are outer  $k$ -planar; this can be shown in a similar manner as [31, Theorem 4] for  $k = 2$ . Hence, we assume the input graph to be biconnected.

► **Theorem 12.** *OUTER  $k$ -PLANARITY can be solved in time  $2^{O(k \log k)} n^{3k+O(1)}$ , that is, OUTER  $k$ -PLANARITY is polynomial-time solvable when  $k$  is fixed.*

Let  $G$  be the input graph, let  $\vec{e} = (u, v)$  be an ordered pair of two distinct vertices of  $G$ , and let  $R$  be a subset of  $V(G) \setminus \{u, v\}$  such that there are at most  $k$  edges between  $R$  and  $L$ , where  $L = V(G) \setminus (\{u, v\} \cup R)$ . Let  $C$  be the set of these edges. Let  $\tau$  be a linear order of

$C$ , let  $c_1, \dots, c_\ell$  denote the vertices of  $C$  in the order given by  $\tau$ , and let  $\chi: C \rightarrow \{0, \dots, k\}$ . Let  $G_{\tau, \vec{e}, R}$  be the graph obtained by adding  $\ell$  vertices  $t_1^\tau, t_2^\tau, \dots, t_\ell^\tau$  to the induced subgraph  $G[\{u, v\} \cup R]$  and by connecting  $t_i^\tau$  and the endpoint of  $c_i$  in  $R$  for every  $i \in [\ell]$ . Then we define a Boolean value  $\mathbf{draw}(\vec{e}, R, \tau, \chi)$  to be **true** if and only if  $G_{\tau, \vec{e}, R}$  admits an outer  $k$ -planar drawing  $D$  with the following properties:

- (P1) the cyclic order of  $D$  contains  $(u, t_1^\tau, t_2^\tau, \dots, t_\ell^\tau, v)$  as a consecutive subsequence, and
- (P2) for every edge  $c_i \in C$ , it holds that  $\chi(c_i) = \text{cr}_D(c_i)$ .

Clearly, the graph  $G$  admits an outer  $k$ -planar drawing if and only if there exists a vertex pair  $\vec{e}$  such that  $\mathbf{draw}(\vec{e}, V(G) \setminus \{u, v\}, f_\emptyset, f_\emptyset) = \mathbf{true}$ , where  $f_\emptyset$  is the empty function.

We evaluate the recurrence as follows. For every base case, namely where  $R = \emptyset$ ,  $\mathbf{draw}(\vec{e}, R, \tau, \chi)$  is **true** since  $C$  is also empty.

When  $R \neq \emptyset$ , we compute  $\mathbf{draw}(\vec{e}, R, \tau, \chi)$  for smaller sets of type  $R$ . In this case, with the same technique as that used in [24, Lemma 6], we can show the following.

► **Lemma 13** (\*). *If  $G_{\tau, \vec{e}, R}$  admits an outer  $k$ -planar drawing  $D$  with properties P1 and P2, there is  $w \in R$  such that vertex pairs  $\{u, w\}$  and  $\{v, w\}$  are pierced by at most  $k$  edges in  $D$ .*

Hence, we can compute  $\mathbf{draw}(\vec{e}, R, \tau, \chi)$  by checking all the ways to split the instance at the vertex  $w$ . Let  $\{R_1, R_2\}$  be a partition of  $R \setminus \{w\}$ , let  $L_1 = V(G_{\tau, \vec{e}, R}) \setminus (\{u, w\} \cup R_1)$  and let  $L_2 = V(G_{\tau, \vec{e}, R}) \setminus (\{v, w\} \cup R_2)$ . For  $i \in [2]$ , let  $C_i$  be the set of edges between  $R_i$  and  $L_i$ , let  $\ell_i = |C_i|$ , let  $\tau_i$  be a linear order of  $C_i$ , and let  $\chi_i: C_i \rightarrow \{0, \dots, k\}$  be a function. We say that  $w, R_1, \tau_1, \chi_1, R_2, \tau_2, \chi_2$  are *consistent* if the following holds ( $\text{cr}_{\tau, \tau_1, \tau_2}$  is defined below):

- there are at most  $k$  edges between  $L_1$  and  $R_1$  and at most  $k$  edges between  $L_2$  and  $R_2$ ,
- for every edge  $c$  between  $L$  and  $R_i$ ,  $\chi_i(c) = \chi(c) - \text{cr}_{\tau, \tau_1, \tau_2}(c)$  holds for  $i \in [2]$ ,
- for every edge  $c$  between  $L$  and  $\{w\}$ ,  $\chi(c) = \text{cr}_{\tau, \tau_1, \tau_2}(c)$ ,
- for every edge  $c$  between  $\{v\}$  and  $R_1$ ,  $\chi_1(c) + \text{cr}_{\tau, \tau_1, \tau_2}(c) \leq k$ ,
- for every edge  $c$  between  $\{u\}$  and  $R_2$ ,  $\chi_2(c) + \text{cr}_{\tau, \tau_1, \tau_2}(c) \leq k$ , and
- for every edge  $c$  between  $R_1$  and  $R_2$ ,  $\chi_1(c) + \chi_2(c) + \text{cr}_{\tau, \tau_1, \tau_2}(c) \leq k$ .

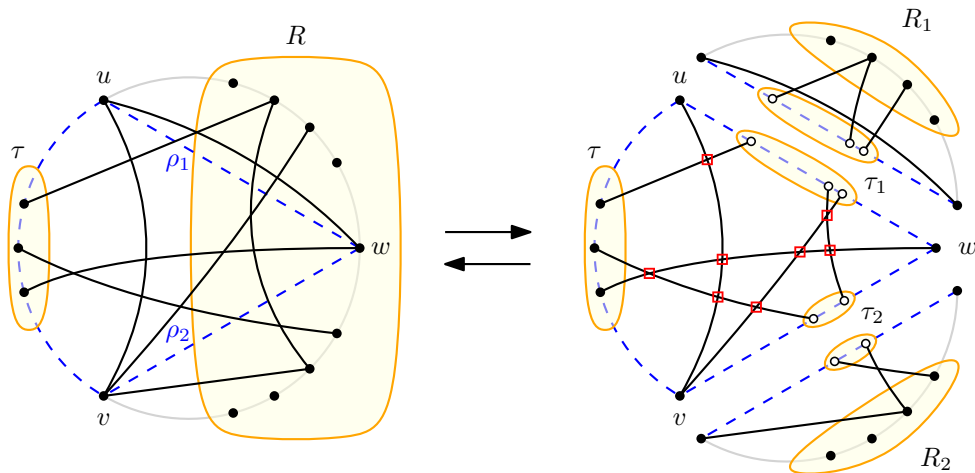
Informally, the value  $\text{cr}_{\tau, \tau_1, \tau_2}(c)$  is the number of crossings on  $c$  inside the “triangle” consisting of  $\{u, v, w\}$ . To define it formally, let us consider a circular drawing  $D_H = (u, t_1^\tau, \dots, t_\ell^\tau, v, t_{\ell_2}^{\tau_2}, \dots, t_{\ell_1}^{\tau_1}, w, t_{\ell_1}^{\tau_1}, \dots, t_1^\tau)$  of a graph  $H$ . For each edge  $c \in (E \cap \{u, v\}) \cup C \cup C_1 \cup C_2$ , the graph  $H$  contains an edge  $f(c)$  defined as follows. If  $c = \{u, v\}$ ,  $f(c)$  simply connects  $u$  and  $v$ . Suppose that  $c$  is incident to exactly one vertex  $x \in \{u, v, w\}$ . This implies that  $c$  is contained in exactly one of  $C$ ,  $C_1$ , and  $C_2$ , which also means that  $c$  is contained in the domain of exactly one  $\tau' \in \{\tau, \tau_1, \tau_2\}$ . Then  $f(c)$  connects  $x$  and  $t_{\tau'(c)}^{\tau'}$ . Otherwise,  $c$  is contained in exactly two of  $C$ ,  $C_1$ , and  $C_2$ , since  $c$  is not contained in  $C \cap C_1 \cap C_2$ . Similarly to the previous case,  $c$  is contained in the domains of distinct  $\tau', \tau'' \in \{\tau, \tau_1, \tau_2\}$ . Then  $f(c)$  connects  $t_{\tau'(c)}^{\tau'}$  and  $t_{\tau''(c)}^{\tau''}$ . Now we define  $\text{cr}_{\tau, \tau_1, \tau_2}(c) = \text{cr}_{D_H}(f(c))$ .

We are ready to state Lemma 14, which formalizes the above idea of splitting an instance  $((u, v), R, \tau, \chi)$  at a vertex  $w$  in  $R$  into two subinstances  $((u, w), R_1, \tau_1, \chi_1)$  and  $((w, v), R_2, \tau_2, \chi_2)$ ; see Figure 4, where some edges are curved for better visualization.

► **Lemma 14.** *For  $\vec{e} = (u, v)$ , it holds that*

$$\mathbf{draw}(\vec{e}, R, \tau, \chi) = \bigvee_{\substack{w, R_1, \tau_1, \chi_1, R_2, \tau_2, \chi_2 \\ \text{consistent}}} \mathbf{draw}((u, w), R_1, \tau_1, \chi_1) \wedge \mathbf{draw}((w, v), R_2, \tau_2, \chi_2).$$

**Proof.** Suppose that  $\mathbf{draw}(\vec{e}, R, \tau, \chi) = \mathbf{true}$ , that is, there is an outer  $k$ -planar drawing  $D = (u, t_1^\tau, \dots, t_\ell^\tau, v, v_1, \dots, v_r)$  of  $G_{\tau, \vec{e}, R}$  that satisfies properties P1 and P2. We assume that edges are drawn as straight-line segments in  $D$ . By Lemma 13 there is a vertex  $w \in R$  for



■ **Figure 4** An image of Lemma 14. The red boxes are the crossings considered in  $\text{cr}_{\tau, \tau_1, \tau_2}$ .

some  $w = v_i$  such that both  $\{u, w\}$  and  $\{v, w\}$  have at most  $k$  piercing edges. The vertices  $u, v, w$  divide the circumference into the three arcs  $\rho_{\bar{u}}, \rho_{\bar{v}}, \rho_{\bar{w}}$ , where  $\rho_{\bar{x}}$  is the arc between vertices other than  $x$  that does not pass through  $x$  for each  $x \in \{u, v, w\}$ . Then, following the line through  $u$  and  $w$ , we can take a curve  $\rho_1$  between  $u$  and  $w$ , inside the circle, such that it crosses exactly the piercing edges, does not pass through any crossing between piercing edges, and separates  $v$  and the segment representing the edge  $\{u, w\}$  (if it exists). We can take a curve  $\rho_2$  between  $v$  and  $w$  similarly. Let  $R_1 = \{v_{i+1}, \dots, v_r\}$  and  $R_2 = \{v_1, \dots, v_{i-1}\}$ . We then define a linear order  $\tau_1$  on the edges between  $R_1$  and  $L_1 := V(G_{\tau, \vec{e}, R}) \setminus (\{u, w\} \cup R_1)$  in such a way that  $\tau_1$  orders those edges in ascending order of the distance between  $u$  and the crossing with the curve  $\rho_1$ . Similarly,  $\tau_2$  is defined in such a way that  $\tau_2$  orders the edges  $R_2$  and  $L_2 := V(G_{\tau, \vec{e}, R}) \setminus (\{v, w\} \cup R_2)$  in ascending order of the distance between  $w$  to the crossing with the curve  $\rho_2$ . If we cut the drawing  $D$  along the curves as Figure 4, the drawing  $D$  can be decomposed into three subdrawings  $D_H, D_1$ , and  $D_2$ :  $D_H$  is the drawing inside the region surrounded by arcs  $\rho_{\bar{w}}, \rho_2$ , and  $\rho_1$ ;  $D_1$  is the drawing inside the region surrounded by arcs  $\rho_{\bar{v}}$  and  $\rho_1$ ;  $D_2$  is the drawing inside the region surrounded by arcs  $\rho_{\bar{u}}$  and  $\rho_2$ . Since each crossing on the edges between  $R_1$  and  $L_1$  is contained in exactly one of  $D_H, D_1, D_2$ , we have  $\chi_1(c) = \chi(c) - \text{cr}_{\tau, \tau_1, \tau_2}(c)$  for each edge  $c$  between  $L$  and  $R_1$ , and we have  $\chi_1(c) + \text{cr}_{\tau, \tau_1, \tau_2}(c) \leq k$  for each edge  $c$  between  $\{v\}$  and  $R_1$ . As  $D_1$  is a circular drawing of  $G_{\tau_1, (u, w), R_1}$  that satisfies P1 and P2, we have  $\text{draw}((u, w), R_1, \tau_1, \chi_1) = \text{true}$ , and similarly, we have  $\text{draw}((w, v), R_2, \tau_2, \chi_2) = \text{true}$ . Since each edge  $c$  between  $R_1$  and  $R_2$  satisfies  $\chi_1(c) + \chi_2(c) + \text{cr}_{\tau, \tau_1, \tau_2}(c) \leq k$ , we conclude that  $w, R_1, \tau_1, \chi_1, R_2, \tau_2, \chi_2$  are consistent.

Suppose that there are consistent  $w, R_1, \tau_1, \chi_1, R_2, \tau_2, \chi_2$  such that  $\text{draw}((u, w), R_1, \tau_1, \chi_1) = \text{draw}((w, v), R_2, \tau_2, \chi_2) = \text{true}$ . Let  $D_1$  and  $D_2$  be circular drawings of  $G_{\tau_1, (u, w), R_1}$  and  $G_{\tau_2, (w, v), R_2}$ , respectively, that satisfy P1 and P2. Let  $\sigma_1 = (w, \dots, u)$  and  $\sigma_2 = (v, \dots, w)$  be the linear orders of  $\{w, u\} \cup R_1$  and  $\{v, w\} \cup R_2$  obtained from  $D_1$  and  $D_2$  by removing the vertices  $t_i^{\tau_1}$  and  $t_j^{\tau_2}$  for  $i \in [\ell_1]$  and  $j \in [\ell_2]$ , respectively. Then we obtain a cyclic order  $\sigma$  of  $G_{\tau, \vec{e}, R}$  by concatenating  $\sigma_2, \sigma_1, (u, t_1^{\tau_1}, \dots, t_{\ell}^{\tau_1}, v)$  in this order, identifying the two occurrences of each of  $w, u, v$  with each other. It is not difficult to see that, by combining  $D_H, D_1$ , and  $D_2$  as in Figure 4, we obtain a drawing  $D$  with linear order  $\sigma$  that satisfies P1 and P2. In other words,  $\text{draw}(\vec{e}, R, \tau, \chi) = \text{true}$ . ◀

Naively, the number of  $R$ 's to consider is  $\Theta(2^n)$ , which does not give an XP-algorithm. However, the following lemma assures that it is not so large.

► **Lemma 15** (\*). *Let  $G$  be a biconnected graph that admits an outer  $k$ -planar drawing  $D$ . Let  $\{u, v\}$  be a pair of distinct vertices of  $G$  that has at most  $k$  piercing edges in  $D$ . Then the number of  $R$ 's such that  $\text{draw}((u, v), R, \tau, \chi) = \text{true}$  for some  $\tau$  and  $\chi$  is at most  $2^{O(k)} m^{k+O(1)}$ , where  $m = |E(G)|$ . Moreover, such  $R$ 's can be enumerated in  $2^{O(k)} m^{k+O(1)}$  time.*

**Proof (sketch).** We show that, given the set of edges piercing  $\{u, v\}$ , there are  $2^{O(k)}$  possibilities for  $R$  that are separated by these piercing edges. Since  $R$  is a union of components in the graph obtained from  $G[V(G) \setminus \{u, v\}]$  by deleting the piercing edges, it suffices to show that there are only  $O(k)$  components in this graph. The proof shares the same underlying idea with Lemma 8, but it is more involved as the maximum degree is no longer bounded. The upper bound can be obtained by considering that there are at most  $k$  piercing edges of  $\{u, v\}$  and the number of components in  $G[V(G) \setminus \{u, v\}]$  is at most  $2k + 3$ . ◀

With Lemma 15 and the fact that  $m = O(\sqrt{k}n)$  [40], the number of combinations of arguments  $\{e, R, \sigma, \chi\}$  to consider is at most

$$n^2 \cdot 2^{O(k)} m^{k+O(1)} \cdot k! \cdot (k+1)^k = 2^{O(k \log k)} n^{k+O(1)}.$$

To compute the value  $\text{draw}(\vec{e}, R, \sigma, \chi)$  as in Lemma 14, we guess at most

$$n \cdot (2^{O(k \log k)} n^{k+O(1)})^2 = 2^{O(k \log k)} n^{2k+O(1)}$$

possible combinations of  $w, R_1, \tau_1, \chi_1, R_2, \tau_2, \chi_2$ . For each guess, checking the consistency takes  $n^{O(1)}$  time. Hence, the total running time to fill the table is  $2^{O(k \log k)} n^{3k+O(1)}$ . This completes the proof of Theorem 12.

## 5.2 NP-Hardness of Approximation

In this subsection, we show an inapproximability result for OUTER  $k$ -PLANARITY even for graphs that are almost trees, whereas trees can be drawn without any crossings.

► **Theorem 16.** *For any fixed  $c \geq 1$ , there is no polynomial-time  $c$ -approximation algorithm for OUTER  $k$ -PLANARITY unless  $P = NP$ , even for graphs with feedback vertex number 1.*

Our proof is by reduction from BANDWIDTH on trees, which is NP-hard to approximate within any constant factor [21]. In other words, given a tree  $T$ , there is no polynomial-time algorithm to distinguish between the cases  $\text{bw}(T) \leq b$  and  $\text{bw}(T) > cb$  for any constant  $c \geq 1$ , unless  $P = NP$ .

Let  $T$  be a tree, and let  $n$  denote  $|V(T)|$ . We construct a graph  $G$  from  $T$  by adding a vertex  $w$  and making it adjacent to all vertices of  $T$ . Clearly,  $G$  has feedback vertex number 1 since  $G[V(G) \setminus \{w\}]$  is a tree.

► **Lemma 17** (\*). *If  $G$  is outer  $k$ -planar, then  $\text{bw}(T) \leq k - 1$ .*

► **Lemma 18** (\*). *Let  $b \geq 1$ . If  $\text{bw}(T) \leq b$ , then  $G$  is outer  $(5b - 5)$ -planar.*

**Proof (sketch).** From a vertex order  $(v_1, \dots, v_n)$  of  $T$  with bandwidth at most  $b$ , we construct a circular drawing  $D$  of  $G$  as  $D = (w, v_1, \dots, v_n)$ . Each edge  $\{w, v_i\}$  incident to  $w$  has at most  $2b - 2$  crossings in  $D$  since these crossing edges lie between vertices that are “close” to  $v_i$ . For other edge  $\{v_i, v_j\} \in E(T)$  with  $i < j$ , it only crosses (1) edges incident to  $w$  and (2)

edges in  $T$ . There are at most  $b - 1$  edges of (1) since the stretch of  $\{v_i, v_j\}$  is at most  $b$ , and at most  $4b - 4$  edges of (2) since these edges lie between vertices that are “close” to  $v_i$  or  $v_j$ . Hence, each edge has at most  $5b - 5$  crossings in total. ◀

Suppose that there is a polynomial-time  $c$ -approximation algorithm  $\mathcal{A}$  for OUTER  $k$ -PLANARITY. Let  $T$  be a tree, and let  $b = \text{bw}(T)$ . By Lemma 18,  $\mathcal{A}$  would output an outer  $5bc$ -planar drawing  $D$  of  $G$ . By Lemma 17,  $D$  can be transformed into a linear order of  $V(T)$  with bandwidth at most  $5bc$ . Thus, we can find a  $5c$ -approximate solution for BANDWIDTH in polynomial time, which is impossible under  $\text{P} \neq \text{NP}$ . This completes the proof of Theorem 16.

### 5.3 XNLP-Hardness

In the proof of Theorem 16, we reduced the gap-version of BANDWIDTH to OUTER  $k$ -PLANARITY. We exploited the gap to accommodate the crossings between edges in the original instance, which may increase the crossing number of each edge by  $O(b)$ . However, if we allow parallel edges, we can reduce from (the exact version of) BANDWIDTH by making the edges incident to  $w$  so thick that we can ignore the  $O(b)$  increase in the crossing numbers. The following theorem is shown by emulating those parallel edges with rigid structures.

► **Theorem 19 (★).** *OUTER  $k$ -PLANARITY is XNLP-hard when parameterized by  $k$ .*

**Proof (sketch).** As in Theorem 11, we give a parameterized logspace reduction from BANDWIDTH on trees. The idea of the reduction is similar to that used in Theorem 16. Instead of connecting  $w$  with each  $v_i$ , we replace each vertex  $v_i$  with a clique path gadget that appears consecutively in any outer  $k$ -planar drawing for some  $k = \Theta(b^4)$  and connect  $w$  with sufficiently many vertices in the gadget. Since there are many edges between  $w$  and each gadget, two adjacent gadgets are placed closely in any outer  $k$ -planar drawing. ◀

## 6 Open Problems

We conclude with a number of problems that we have left open in this paper.

- Is ONE-SIDED  $k$ -PLANARITY NP-hard?
- We conjecture that OUTER  $k$ -PLANARITY is XALP-complete (see [12] for the definition).
- Can we extend the algorithm for TWO-SIDED  $k$ -PLANARITY to obtain an XP-algorithm for  $\ell$ -layer  $k$ -planarity parameterized by  $\ell + k$ ?
- Another way to extend  $k$ -planarity is to consider *min- $k$ -planarity*, which is also called *weak  $k$ -planarity* [9, 15]. In a min- $k$ -planar drawing, in every crossing, at least one of the two edges must have at most  $k$  crossings. Can 2-layer min- $k$ -planar graphs and outer min- $k$ -planar graphs be recognized by XP-algorithms with respect to  $k$ ?
- TWO-SIDED  $k$ -PLANARITY can be seen as a restricted version of OUTER  $k$ -PLANARITY for bipartite graphs where the vertices of the two sets of the bipartition must not interleave in the cyclic vertex order. This can be generalized as follows: For  $k \geq 3$ , can we efficiently recognize  $k$ -partite graphs that admit a  $k$ -planar straight-line drawing on the regular  $k$ -gon? A related question has been investigated for fixed-order book embedding [1].

---

### References

- 1 Akanksha Agrawal, Sergio Cabello, Michael Kaufmann, Saket Saurabh, Roohani Sharma, Yushi Uno, and Alexander Wolff. Eliminating crossings in ordered graphs. In Hans Bodlaender, editor, *19th Scand. Symp. Algorithm Theory (SWAT)*, volume 294 of *LIPIcs*, pages 1:1–1:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.SWAT.2024.1.

- 2 Oswin Aichholzer, Johannes Obenaus, Joachim Orthaber, Rosna Paul, Patrick Schnider, Raphael Steiner, Tim Taubner, and Birgit Vogtenhuber. Edge partitions of complete geometric graphs. In Xavier Goaoc and Michael Kerber, editors, *38th Int. Symp. Comput. Geom. (SoCG)*, volume 224 of *LIPIcs*, pages 6:1–6:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.SocG.2022.6.
- 3 Patrizio Angelini, Giordano Da Lozzo, Henry Förster, and Thomas Schneck. 2-Layer  $k$ -planar graphs: Density, crossing lemma, relationships and pathwidth. *The Computer Journal*, 67(3):1005–1016, 2023. doi:10.1093/comjnl/bxad038.
- 4 Christopher Auer, Christian Bachmaier, Franz J. Brandenburg, Andreas Gleißner, Kathrin Hanauer, Daniel Neuwirth, and Josef Reislhuber. Outer 1-planar graphs. *Algorithmica*, 74(4):1293–1320, 2016. doi:10.1007/S00453-015-0002-1.
- 5 Michael Bannister, Sergio Cabello, and David Eppstein. Parameterized complexity of 1-planarity. *Journal of Graph Algorithms and Applications*, 22(1):23–49, 2018. doi:10.7155/jgaa.00457.
- 6 Michael Bannister and David Eppstein. Crossing minimization for 1-page and 2-page drawings of graphs with bounded treewidth. *Journal of Graph Algorithms and Applications*, 22(4):577–606, 2018. doi:10.7155/jgaa.00479.
- 7 Michael Baur and Ulrik Brandes. Crossing reduction in circular layouts. In Juraj Hromkovic, Manfred Nagl, and Bernhard Westfechtel, editors, *30th Int. Workshop Graph-Theoretic Concepts Comput. Sci. (WG)*, volume 3353 of *LNCS*, pages 332–343. Springer, 2004. doi:10.1007/978-3-540-30559-0\_28.
- 8 Sujoy Bhore, Robert Ganian, Fabrizio Montecchiani, and Martin Nöllenburg. Parameterized algorithms for book embedding problems. *Journal of Graph Algorithms and Applications*, 24(4):603–620, 2020. doi:10.7155/jgaa.00526.
- 9 Carla Binucci, Aaron Büngener, Giuseppe Di Battista, Walter Didimo, Vida Dujmovic, Seok-Hee Hong, Michael Kaufmann, Giuseppe Liotta, Pat Morin, and Alessandra Tappini. Min- $k$ -planar drawings of graphs. *Journal of Graph Algorithms and Applications*, 28(2):1–35, 2024. doi:10.7155/JGAA.V28I2.2925.
- 10 Václav Blažej, Boris Klemz, Felix Klesen, Marie Diana Sieper, Alexander Wolff, and Johannes Zink. Constrained and ordered level planarity parameterized by the number of levels. In Wolfgang Mulzer and Jeff M. Phillips, editors, *40th Annu. Sympos. Comput. Geom. (SoCG'24)*, volume 293 of *LIPIcs*, pages 21:1–16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.SocG.2024.20.
- 11 Hans L. Bodlaender. Parameterized complexity of bandwidth of caterpillars and weighted path emulation. In Łukasz Kowalik, Michał Pilipczuk, and Paweł Rzażewski, editors, *Graph-Theoretic Concepts Comput. Sci. (WG)*, volume 12911 of *LNCS*, pages 15–27. Springer, 2021. doi:10.1007/978-3-030-86838-3\_2.
- 12 Hans L. Bodlaender, Carla Groenland, Hugo Jacob, Marcin Pilipczuk, and Michał Pilipczuk. On the complexity of problems on tree-structured graphs. In Holger Dell and Jesper Nederlof, editors, *17th Int. Symp. Paramet. & Exact Comput. (IPEC)*, volume 249 of *LIPIcs*, pages 6:1–6:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.IPEC.2022.6.
- 13 Hans L. Bodlaender, Carla Groenland, Jesper Nederlof, and Céline M. F. Swennenhuis. Parameterized problems complete for nondeterministic FPT time and logarithmic space. In *62nd IEEE Ann. Symp. Foundat. Comput. Sci. (FOCS)*, pages 193–204, 2021. doi:10.1109/FOCS52979.2021.00027.
- 14 Sergio Cabello and Bojan Mohar. Adding one edge to planar graphs makes crossing number and 1-planarity hard. *SIAM Journal on Computing*, 42(5):1803–1829, 2013. doi:10.1137/120872310.
- 15 Rutger Campbell, Katie Clinch, Marc Distel, J. Pascal Gollin, Kevin Hendrey, Robert Hickingbotham, Tony Huynh, Freddie Illingworth, Youri Tamitegama, Jane Tan, and David R. Wood. Product structure of graph classes with bounded treewidth. *Combinatorics, Probability and Computing*, 33(3):351–376, 2024. doi:10.1017/S0963548323000457.



- 16 Steven Chaplick, Myroslav Kryven, Giuseppe Liotta, Andre Löffler, and Alexander Wolff. Beyond outerplanarity. In Fabrizio Frati and Kwan-Liu Ma, editors, *25th Int. Symp. Graph Drawing & Network Vis. (GD)*, volume 10692 of *LNCS*, pages 546–559. Springer, 2018. doi:10.1007/978-3-319-73915-1\_42.
- 17 Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.
- 18 Walter Didimo, Giuseppe Liotta, and Fabrizio Montecchiani. A survey on graph drawing beyond planarity. *ACM Comput. Surv.*, 52(1):4:1–4:37, 2019. doi:10.1145/3301281.
- 19 Alexander Dobler. A note on the complexity of one-sided crossing minimization of trees, 2023. arXiv. doi:10.48550/arXiv.2306.15339.
- 20 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.
- 21 Chandan Dubey, Uriel Feige, and Walter Unger. Hardness results for approximating the bandwidth. *Journal of Computer and System Sciences*, 77(1):62–90, 2011. Celebrating Karp’s Kyoto Prize. doi:10.1016/j.jcss.2010.06.006.
- 22 Vida Dujmović, Seok-Hee Hong, Michael Kaufmann, János Pach, and Henry Förster. Beyond-planar graphs: Models, structures and geometric representations (Dagstuhl seminar 24062). *Dagstuhl Reports*, 14(2):71–94, 2024. doi:10.4230/DagRep.14.2.71.
- 23 Michael Elberfeld, Christoph Stockhusen, and Till Tantau. On the space and circuit complexity of parameterized problems: Classes and completeness. *Algorithmica*, 71(3):661–701, 2015. doi:10.1007/s00453-014-9944-y.
- 24 Oksana Firman, Grzegorz Gutowski, Myroslav Kryven, Yuto Okada, and Alexander Wolff. Bounding the treewidth of outer  $k$ -planar graphs via triangulations. In Stefan Felsner and Karsten Klein, editors, *32nd Int. Symp. Graph Drawing & Network Vis. (GD)*, volume 320 of *LIPICs*, pages 14:1–14:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICs.GD.2024.14.
- 25 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006. doi:10.1007/3-540-29953-X.
- 26 Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- 27 Michael R. Garey and David S. Johnson. Crossing number is NP-complete. *SIAM Journal on Algebraic Discrete Methods*, 4(3):312–316, 1983. doi:10.1137/0604033.
- 28 Alexander Grigoriev and Hans L. Bodlaender. Algorithms for graphs embeddable with few crossings per edge. *Algorithmica*, 49(1):1–11, 2007. doi:10.1007/S00453-007-0010-X.
- 29 Martin Grohe. Computing crossing numbers in quadratic time. *Journal of Computer and System Sciences*, 68(2):285–302, 2004. doi:10.1016/j.jcss.2003.07.008.
- 30 Seok-Hee Hong, Peter Eades, Naoki Katoh, Giuseppe Liotta, Pascal Schweitzer, and Yusuke Suzuki. A linear-time algorithm for testing outer-1-planarity. *Algorithmica*, 72(4):1033–1054, 2015. doi:10.1007/S00453-014-9890-8.
- 31 Seok-Hee Hong and Hiroshi Nagamochi. A linear-time algorithm for testing full outer-2-planarity. *Discrete Applied Mathematics*, 255:234–257, 2019. doi:10.1016/j.dam.2018.08.018.
- 32 Paul C. Kainen. The book thickness of a graph. II. In *20th Southeastern Conf. Combin., Graph Theory, & Comput. (Boca Raton, FL, 1989)*, volume 71, pages 127–132, 1990.
- 33 Ken-ichi Kawarabayashi and Bruce Reed. Computing crossing number in linear time. In *39th Ann. ACM Symp. Theory Comput. (STOC)*, pages 382–390, 2007. doi:10.1145/1250790.1250848.
- 34 Yasuaki Kobayashi, Hiromu Ohtsuka, and Hisao Tamaki. An improved fixed-parameter algorithm for one-page crossing minimization. In Daniel Lokshtanov and Naomi Nishimura, editors, *12th Int. Symp. Paramet. & Exact Comput. (IPEC)*, volume 89 of *LIPICs*, pages 25:1–25:12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPICs.IPEC.2017.25.

- 35 Yasuaki Kobayashi and Hisao Tamaki. A fast and simple subexponential fixed parameter algorithm for one-sided crossing minimization. *Algorithmica*, 72:778–790, 2015. doi:10.1007/s00453-014-9872-x.
- 36 Yasuaki Kobayashi and Hisao Tamaki. A faster fixed parameter algorithm for two-layer crossing minimization. *Information Processing Letters*, 116(9):547–549, 2016. doi:10.1016/j.ipl.2016.04.012.
- 37 Yunlong Liu, Jie Chen, and Jingui Huang. Parameterized algorithms for fixed-order book drawing with bounded number of crossings per edge. In Weili Wu and Zhongnan Zhang, editors, *Proc. 14th Int. Conf. Combin. Optim. Appl. (COCOA)*, volume 12577 of *LNCS*, pages 562–576. Springer, 2020. doi:10.1007/978-3-030-64843-5\_38.
- 38 Yunlong Liu, Jie Chen, Jingui Huang, and Jianxin Wang. On parameterized algorithms for fixed-order book thickness with respect to the pathwidth of the vertex ordering. *Theor. Comput. Sci.*, 873:16–24, 2021. doi:10.1016/j.tcs.2021.04.021.
- 39 Xavier Muñoz, Walter Unger, and Imrich Vrt'o. One sided crossing minimization is NP-hard for sparse graphs. In Petra Mutzel, Michael Jünger, and Sebastian Leipert, editors, *9th Int. Symp. Graph Drawing (GD)*, volume 2265 of *LNCS*, pages 115–123. Springer, 2001. doi:10.1007/3-540-45848-4\_10.
- 40 János Pach and Géza Tóth. Graphs drawn with few crossings per edge. *Combinatorica*, 17(3):427–439, 1997. doi:10.1007/BF01215922.
- 41 Michał Pilipczuk and Marcin Wrochna. On space efficiency of algorithms working on structural decompositions of graphs. *ACM Trans. Comput. Theory*, 9(4), 2018. doi:10.1145/3154856.
- 42 Helen C. Purchase, Christopher Pilcher, and Beryl Plimmer. Graph drawing aesthetics – created by users, not algorithms. *IEEE Transactions on Visualization and Computer Graphics*, 18(1):81–92, 2012. doi:10.1109/TVCG.2010.269.
- 43 James B. Saxe. Dynamic-programming algorithms for recognizing small-bandwidth graphs in polynomial time. *SIAM J. Algebraic Discret. Methods*, 1(4):363–369, 1980. doi:10.1137/0601042.
- 44 Marcus Schaefer. The graph crossing number and its variants: A survey. *Electronic Journal of Combinatorics*, DS21, 2024. doi:10.37236/2713.
- 45 Farhad Shahrokhi, Ondrej Sýkora, László A. Székely, and Imrich Vrt'o. On bipartite drawings and the linear arrangement problem. *SIAM Journal on Computing*, 30(6):1773–1789, 2001. doi:10.1137/S0097539797331671.
- 46 Kozo Sugiyama, Shojiro Tagawa, and Mitsuhiro Toda. Methods for visual understanding of hierarchical system structures. *IEEE Trans. Syst. Man Cybern.*, 11(2):109–125, 1981. doi:10.1109/TSMC.1981.4308636.
- 47 John C. Urschel and Jake Wellens. Testing gap  $k$ -planarity is NP-complete. *Information Processing Letters*, 169:106083, 2021. doi:10.1016/j.ipl.2020.106083.
- 48 Meirav Zehavi. Parameterized analysis and crossing minimization problems. *Computer Science Review*, 45:100490, 2022. doi:10.1016/j.cosrev.2022.100490.