Outerplanar and Forest Storyplans

Jiří Fiala
Charles University, Prague, Czech Republic

Oksana Firman
Institut für Informatik, Universität Würzburg, Germany

Giuseppe Liotta
Università degli Studi di Perugia, Perugia, Italy

Alexander Wolff
Institut für Informatik, Universität Würzburg, Germany

Johannes Zink
TUM School of Computation, Information and Technology, Technische Universität München, Germany

Abstract

We study the problem of gradually representing a complex graph as a sequence of drawings of small subgraphs whose union is the complex graph. The sequence of drawings is called *storyplan*, and each drawing in the sequence is called a *frame*. In an (outer)planar storyplan, every frame is (outer)planar; in a forest storyplan, every frame is acyclic. Binucci, Di Giacomo, Lenhart, Liotta, Montecchiani, Nöllenburg, and Symvonis [JCSS 2024] proved that every graph of treewidth at most 3 admits a planar storyplan and that deciding whether a given graph admits a *planar* storyplan is NP-complete. They also presented two FPT algorithms, one parameterized with respect to the vertex cover number and one with respect to the feedback edge set number of the input graph.

We first prove that deciding whether a given graph admits an outerplanar storyplan (or a forest storyplan) is NP-complete. Then, we show that the FPT algorithms of Binucci et al. also work for our problem variants with small modifications. We identify graph families that admit outerplanar and forest storyplans and graph families for which such storyplans do not always exist. In the affirmative case, we present efficient algorithms that produce storyplans with straight-line edges.

Keywords and phrases Planar storyplans, outerplanar storyplans, forest storyplans, NP-hard, FPT

Digital Object Identifier 10.57717/cgt.v5i1.103

Related Version A preliminary version of this work has appeared in Proc. 49th International Conference on Current Trends in Theory & Practice of Computer Science (SOFSEM'24), Springer-Verlag, 2024 [20].

Funding *Giuseppe Liotta*: Supported by MUR PON project ARS01 00540 and by MUR PRIN project 2022TS4Y3N.

1 Introduction

A possible approach to the visual exploration of large and complex networks is to gradually display them by showing a sequence of *frames*, where each frame contains the drawing of a portion of the graph. When going from one frame to the next, some vertices and edges appear while others disappear. To preserve the mental map, the geometric representation of vertices and edges that are shared by two consecutive frames must remain the same. Informally speaking, a *storyplan* for a graph consists of a sequence of frames such that every vertex and edge of the graph appears in at least one frame. Moreover, there is a consistency requirement (as for the labels in a zoomable digital map [2, 3]): once a vertex disappears, it may not re-appear. Hence, after a vertex appears, it remains visible until all its incident

© Jiří Fiala, Oksana Firman, Giuseppe Liotta, Alexander Wolff, and Johannes Zink
licensed under Creative Commons License CC-BY 4.0

edges are represented; then it disappears in the transition to the next frame. See Figure 1 for a storyplan.

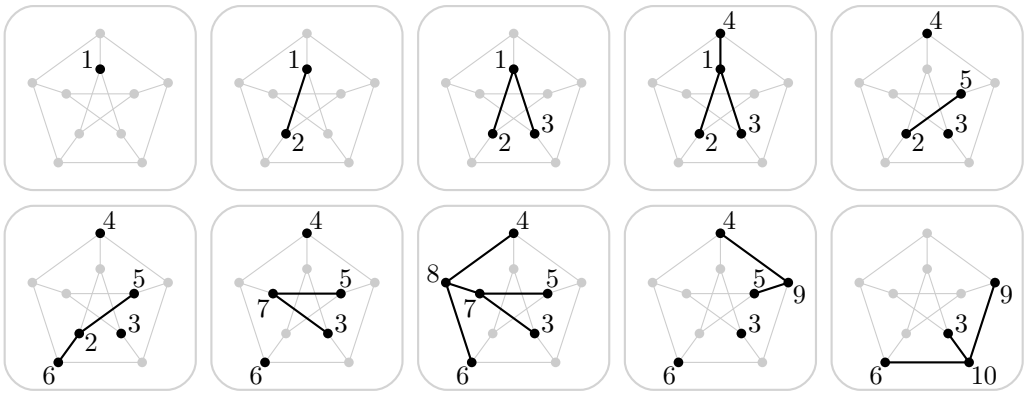


Figure 1 A forest storyplan of the Petersen graph.

Since edge crossings are a natural obstacle to the readability of a graph layout [16], Binucci, Di Giacomo, Lenhart, Liotta, Montecchiani, Nöllenburg, and Symvonis [5, 6] introduced and studied the *planar storyplan problem* that asks whether a given graph admits a storyplan such that every frame is a crossing-free drawing and in every frame a single new vertex appears. We call this problem Planar Storyplan. Binucci et al. showed that Planar Storyplan is NP-complete in general and fixed-parameter tractable with respect to the vertex cover number and with respect to the feedback edge set number. They also proved that every graph of treewidth at most 3 admits a planar storyplan.

Motivated by the research of Binucci et al., we forward the idea of representing a graph with a storyplan such that each frame is a drawing whose visual inspection is as simple as possible. Specifically, we study the *outerplanar storyplan problem* and the *forest storyplan problem*, which are defined analogously to the planar storyplan problem (see Definition 1 below). We call the problems Outerplanar Storyplan and Forest Storyplan, respectively. We let the classes of graphs that admit planar, outerplanar and forest storyplans be denoted by $\mathcal{G}_{\text{planar}}$, $\mathcal{G}_{\text{outerpl}}$, and $\mathcal{G}_{\text{forest}}$, respectively. Clearly, $\mathcal{G}_{\text{forest}} \subseteq \mathcal{G}_{\text{outerpl}} \subseteq \mathcal{G}_{\text{planar}} \subseteq \mathcal{G}$, where $\mathcal{G}$ is the class of all graphs. To further simplify visual inspection, our algorithms draw all frames with straight-line edges. Following Dobler, Holzmüller, and Nöllenburg [19], we call storyplans with this property *geometric storyplans*. They constructed a graph that admits a planar storyplan but no geometric planar storyplan. By modifying the NP-hardness proof of Binucci et al., Dobler et al. showed that the geometric version of Planar Storyplan is also NP-hard.

Beside the work of Binucci et al., our research relates to the graph drawing literature that assumes either dynamic or streaming models (see, e.g., [1, 4, 9, 11, 12]) and to recent work about graph stories (see, e.g., [14, 15, 8, 7]). The key difference to our work is that these papers (except [6]) assume that the order of the vertices is given as part of the input. We now summarize our contribution, using *$\triangle$-free* as shorthand for triangle-free.

- We establish the chain of strict containment relations $\mathcal{G}_{\text{forest}} \subsetneq \mathcal{G}_{\text{outerpl}} \subsetneq \mathcal{G}_{\text{planar}} \subsetneq \mathcal{G}$ (see Figure 2) by showing that
  - there is a $\triangle$-free 6-regular graph that does not admit a planar storyplan;
  - there is a K_4-free 4-regular planar graph that (trivially) admits a planar storyplan, but does not admit an outerplanar storyplan; and

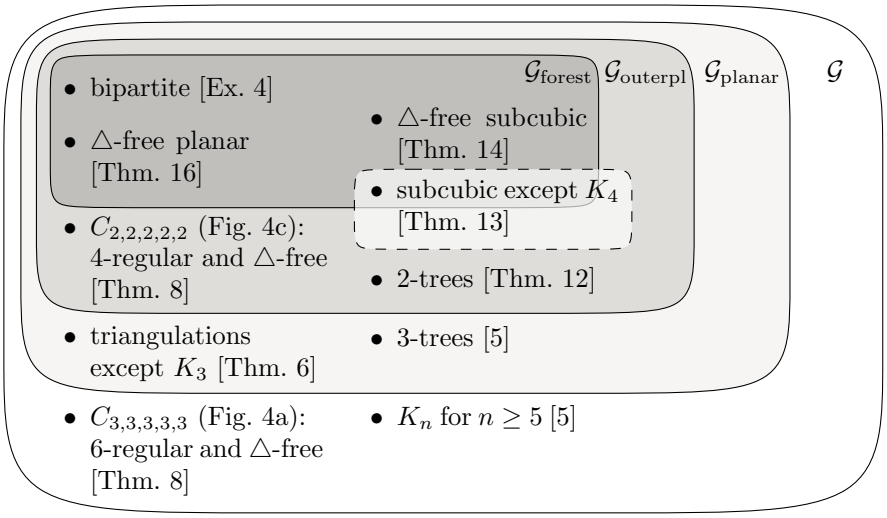


Figure 2 The relationship $\mathcal{G}_{\text{forest}} \subsetneq \mathcal{G}_{\text{outerpl}} \subsetneq \mathcal{G}_{\text{planar}} \subsetneq \mathcal{G}$ is strict: we list graphs and graph classes that are contained in one class but not in the other. For example, all triangulations admit a planar storyplan but not an outerplanar storyplan with the exception of K_3, which, like all 2-trees, admits an outerplanar storyplan. Note that subcubic graphs except K_4 admit an outerplanar storyplan and some also a forest storyplan. Existing results are marked with reference [6], while new results show the name of the theorem/example. For the definition of $C_{2,2,2,2,2}$ and $C_{3,3,3,3,3}$, see Theorem 8.

  - there is a △-free 4-regular (nonplanar) graph that admits an outerplanar storyplan, but does not admit a forest storyplan.
- Recall that a *triangulation* is a maximal planar graph; it admits a planar drawing where every face (including the outer one) is a triangle. We show that no triangulation (except for K_3) admits an outerplanar storyplan; see Section 3.
- We show that both problems Outerplanar Storyplan and Forest Storyplan are NP-complete; Section 4.
- Further we modify the FPT algorithms of Binucci et al. [6] for Planar Storyplan in order to obtain FPT algorithms for Outerplanar Storyplan and for Forest Storyplan; see Section 5.
- On the positive side, we exhibit two graph classes that admit geometric outerplanar storyplans. These are subcubic graphs (except K_4) and partial 2-trees – the latter is the class of graphs of treewidth at most 2; it contains all outerplanar graphs as a proper subclass. Both constructions can be performed in linear time; see Section 6. For subcubic graphs, every frame of the resulting storyplans contains at most five edges.
- A graph must be △-free in order to admit a forest storyplan. We show that △-free subcubic graphs (as the Petersen graph in Figure 1), and △-free planar graphs admit geometric forest storyplans (which we can compute in linear and polynomial time, respectively); see Section 7.

We start with some preliminaries in Section 2 and close with open problems in Section 8.

2 Preliminaries

Given positive integers m, n, we use $[m, n]$ as shorthand for $\{m, m+1, \ldots, n\}$, and we use $[n]$ as shorthand for $[1, n]$. Given a graph G, we let $V(G)$ denote the vertex set of G, and we

let $E(G) \subseteq \binom{V(G)}{2}$ denote the edge set of G.

Our definitions of a planar, an outerplanar, and a forest storyplan are based on the definition of a planar storyplan of Binucci et al. [6].

▶ **Definition 1.** *A planar storyplan $\mathcal{S} = \langle \alpha, (D_i)_{i\in[n]} \rangle$ of G is a pair defined as follows. The first element is a bijection $\alpha\colon V(G) \to [n]$ that represents a total order of the vertices of G. For a vertex $v \in V(G)$, let $\omega(v) = \max_{u\in N[v]} \alpha(u)$, where $N[v]$, the closed neighborhood of v, is the set containing v and its neighbors. The interval $[\alpha(v), \omega(v)]$ is the lifespan of v. We say that v appears at step $\alpha(v)$, is visible at step i for each $i \in [\alpha(v), \omega(v)]$, and disappears at step $\omega(v)+1$. Note that a vertex disappears only when all its neighbors have appeared. The second element of $\mathcal{S}$ is a sequence of drawings $(D_i)_{i\in[n]}$, called frames of $\mathcal{S}$, such that, for $i \in [n]$: (i) D_i is a drawing of the graph G_i induced by the vertices visible at step i, (ii) D_i is planar, (iii) the point representing a vertex v is the same over all drawings that contain v, and (iv) the curve representing an edge e is the same over all drawings that contain e.*

We emphasize that though in our definition of storyplans we allow that edges can be represented by curves, our constructions use only straight-line edges. For an *outerplanar storyplan* and a *forest storyplan*, we strengthen requirement (ii) to D_i being outerplanar and D_i being a crossing-free drawing of a forest, respectively. In what follows, we will sometimes use a slight variant of Definition 1, in which we enrich the sequence $(D_i)_{i\in[n]}$ of frames by explicitly representing the portions of the drawings that consecutive frames have in common. More precisely, for $i \in [n-1]$, let $D'_i = D_i \cap D_{i+1}$. Then, a storyplan is a sequence of drawings $\langle D_1, D'_1, \ldots, D_{n-1}, D'_{n-1}, D_n \rangle$, where in each step $i < n$, we first introduce a vertex (in D_i) and then remove all *completed* vertices (in D'_i), that is, the vertices that disappear in the next step. For $i \in [n-1]$, we define (similarly to D'_i) G'_i as the graph induced by the vertices of the graphs G_i and G_{i+1}. We now list some useful observations.

▶ **Property 2.** *If a graph G admits a planar, an outerplanar, or a forest storyplan, then the same holds for any subgraph of G. Conversely, if a graph G does not admit a planar, an outerplanar, or a forest storyplan, then the same holds for all supergraphs of G.*

▶ **Lemma 3** ([6])**.** *Let $K_{a,b} = (A \cup B, E)$ be a complete bipartite graph with $a = |A|$, $b = |B|$, and $3 \le a \le b$. Let $\mathcal{S} = \langle \alpha, \{D_i\}_{i\in[a+b]} \rangle$ be a planar storyplan of $K_{a,b}$. Exactly one of A and B is such that all its vertices are visible in some frame $i \in [a+b]$.*

▶ **Example 4** (Bipartite graphs)**.** Every bipartite graph admits a forest storyplan: first add all vertices of one set of the bipartition and then, one by one, the vertices of the other set. Note that each vertex of the second set is visible in only one frame.

Now we modify Lemma 3 in order to obtain a property for outerplanar (forest) storyplans of complete bipartite graphs. The proof of Lemma 5 is analogous to the proof of Lemma 3 (see [6]).

▶ **Lemma 5.** *Let $K_{a,b} = (A \cup B, E)$ be a complete bipartite graph with $a = |A|$, $b = |B|$, and $3 \le a \le b$ ($2 \le a \le b$). Let $\mathcal{S} = \langle \alpha, \{D_i\}_{i\in[a+b]} \rangle$ be an outerplanar (forest) storyplan of $K_{a,b}$. One of partition A and B is such that all its vertices are visible at some $i \in [a+b]$ and no two vertices from the other partition are visible at any step $j \in [a+b]$.*

Proof. First we show that there exists a frame, where A or B is completely visible. Recall that $\alpha(v) \in [n]$ is the step where a vertex v appears and $\omega(v) \in [n]$ with $\alpha(v) \le \omega(v)$ is the step where v disappears. Let i be such that D_i contains the largest number t of vertices of A over all frames of $\mathcal{S}$. If $t = a$, we are done. If $t < a$, there exist two vertices u, v of A such

that $\omega(u) < \alpha(v)$. Note that all vertices in B are adjacent to u, and hence they all appear at some step smaller than or equal to $\omega(u)$. On the other hand, since all vertices in B are adjacent to v as well, they cannot disappear before $\alpha(v) + 1$. It follows that all vertices of B are visible at step $\omega(u)$.

Now we show that no two vertices from one of the partitions A and B are visible at any frame $j \in [a + b]$. Consider the interval $I = [s, t] \subseteq [a + b]$ of maximal length such that the vertices of A or B, say A, are all visible. As we have already shown, I is not empty. Let k be one of the steps that contains the largest number b' of vertices of B. Observe that, since I is maximal, one vertex of A appears at s. Therefore, any vertex of B visible at a step smaller than s is visible also at step s. Similarly, by the maximality of I, one vertex of A disappears at step $t + 1$, therefore any vertex of B visible at a step greater than t is visible also at step t. Consequently, we can assume that $k \in I$, and we can conclude that $b' \leq 1$ since it is not possible to have three (two) vertices from one partition and two vertices from the other partition at any frame of the outerplanar (forest) storyplan $\mathcal{S}$. ◀

Note that if one of the partition A or B has exactly two vertices, at most two vertices of the other partition are visible at some frame of an outerplanar storyplan.

3 Separation of graph classes

Trivially, triangulations admit planar storyplans, but as we show now, no triangulation (except for K_3) admits an outerplanar storyplan.

▶ **Theorem 6.** *No triangulation (except for K_3) admits an outerplanar storyplan.*

Proof. For a triangulation, the closed neighborhood of each vertex induces a wheel, which is not outerplanar. For the first vertex that disappears according to a given storyplan, however, its whole closed neighborhood, which is not outerplanar, must be visible. ◀

▶ **Example 7** (Platonic graphs)**.** According to Theorem 6, the tetrahedron, the octahedron, and the icosahedron do not admit outerplanar storyplans because they are triangulations. The cube is bipartite; hence, it admits a forest storyplan due to Example 4. The dodecahedron admits a forest storyplan, see Figure 3. Indeed, the dodecahedron is $\triangle$-free and cubic, and we will show later in Theorem 17 that these conditions are sufficient for the existence of a forest storyplan.

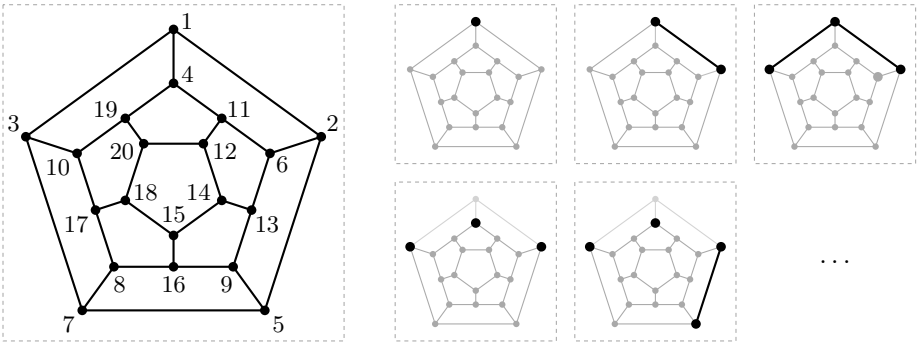
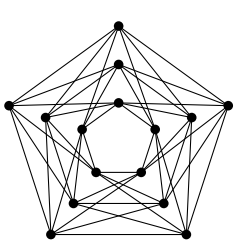
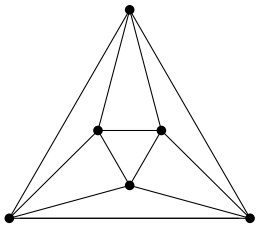


Figure 3 The dodecahedron with a vertex numbering (left) that corresponds to a forest storyplan (started to the right).

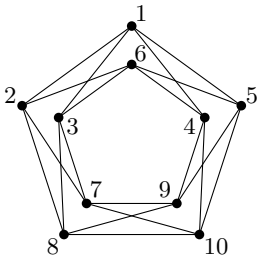
We now separate the graph classes $\mathcal{G}_{\text{forest}}$, $\mathcal{G}_{\text{outerpl}}$, $\mathcal{G}_{\text{planar}}$, and $\mathcal{G}$; see Figure 2.



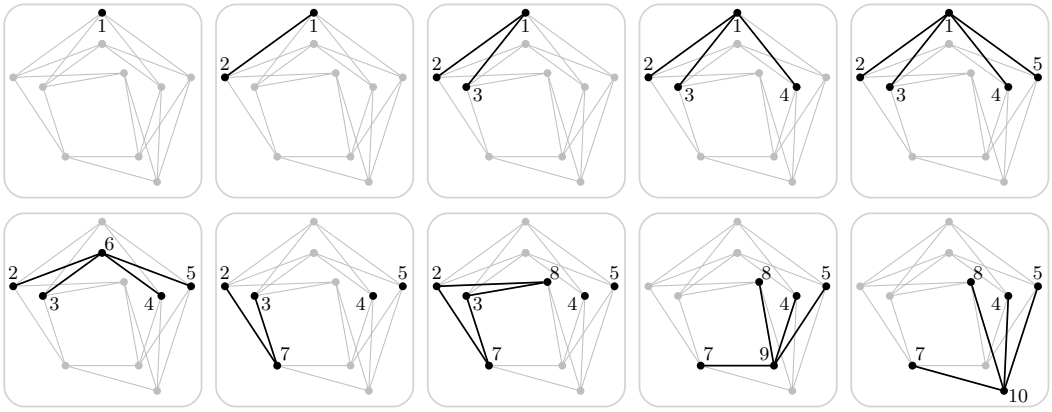
(a) $C_{3,3,3,3,3}$



(b) the octahedron graph



(c) $C_{2,2,2,2,2}$



(d) an outerplanar storyplan for $C_{2,2,2,2,2}$

Figure 4 Three graphs from the proof of Theorem 8. The graph in (a) is $\triangle$-free and does not admit any planar storyplan. The octahedron graph in (b) does not admit any outerplanar storyplan. The graph in (c) is $\triangle$-free and does not admit any forest storyplan, (d) the vertex numbering corresponds to an outerplanar storyplan – if vertex 8 is placed near the position of vertex 6.

▶ **Theorem 8.** *The following statements hold:*

1. *There is a $\triangle$-free 6-regular graph that does not admit a planar storyplan; hence $\mathcal{G}_{\text{planar}} \subsetneq \mathcal{G}$.*
2. *There is a K_4-free 4-regular planar graph that does not admit an outerplanar storyplan; hence $\mathcal{G}_{\text{outerpl}} \subsetneq \mathcal{G}_{\text{planar}}$.*
3. *There is a $\triangle$-free 4-regular (nonplanar) graph that admits an outerplanar storyplan, but does not admit a forest storyplan; hence $\mathcal{G}_{\text{forest}} \subsetneq \mathcal{G}_{\text{outerpl}}$.*

Proof. 1. The graph $C_{3,3,3,3,3}$ (see Figure 4a) is $\triangle$-free and 6-regular, but does not admit a planar storyplan as we will now show. For brevity, let $G = C_{3,3,3,3,3}$, and let $V_1 \cup \cdots \cup V_5$ be the partition of the vertex set of G into independent sets of size 3. Note that, for $i \in [5]$, $G[V_i \cup V_{(i \bmod 5)+1}]$ is isomorphic to $K_{3,3}$. For $K_{3,3} = G[V_1 \cup V_2]$, we know by Lemma 3 that, in any planar storyplan, either all vertices of V_1 or all vertices of V_2 are shown simultaneously, say, those of V_1. This implies, again by Lemma 3, that the vertices of V_2, but also the vertices of V_5 are *not* shown simultaneously. This, in turn, means that the vertices of V_3 *are* shown simultaneously, and the vertices of V_4 are also shown simultaneously. Neither the vertices of V_3 can disappear before the vertices of V_4 have appeared and vice versa. Thus, there must be a frame with a drawing of the non-planar graph $G[V_3 \cup V_4] = K_{3,3}$.

2. Observe that the octahedron (see Figure 4b) is planar, 4-regular, and K_4-free, but does not admit an outerplanar storyplan due to Example 7.

3. The graph $C_{2,2,2,2,2}$ (see Figure 4c) is $\triangle$-free and 4-regular, but does not admit a forest storyplan. The proof is analogous to the one for the graph $C_{3,3,3,3,3}$. There needs to be a frame with a drawing of $K_{2,2}$, which is not a tree. On the other hand, the order of the vertices shown in Figure 4c yields an outerplanar storyplan. Note that we cannot use the vertex positions exactly as in the figure, but if we place vertex 8 near the position of vertex 6 (which will have disappeared by then), every frame is crossing-free (see Figure 4d). ◀

4 Computational complexity

In this section, we show that both problems Forest Storyplan and Outerplanar Storyplan are NP-hard. To this end, we reduce from Positive One-In-Three 3-SAT in such a way that, if the given 3-SAT formula is a yes-instance, then the graph that we construct admits a forest storyplan (and hence an outerplanar storyplan), whereas if the 3-SAT formula is a no-instance, then the graph that we construct does not even admit an outerplanar storyplan (let alone a forest storyplan).

In One-In-Three 3-SAT one is given a Boolean formula in conjunctive normal form with at most three literals per clause and the task is to decide whether there exists a variable assignment such that in every clause exactly one literal is assigned to be *true*. By Schaefer's dichotomy theorem [21], One-In-Three 3-SAT is NP-hard even if no negative literals are admitted; the problem is then called Positive One-In-Three 3-SAT.

Given an instance F of Positive One-In-Three 3-SAT, we construct in polynomial time a graph G_F such that G_F admits a forest storyplan, when F is a yes-instance, and it does not admit even an outerplanar storyplan, when F is a no-instance. We may assume that F does not contain a clause with two occurrences of the same variable. (If there was such a clause of size 2, then F would be a no-instance. Otherwise, in a clause of size 3, the variable that occurs twice would have to be assigned to *false*, and we could remove it from the instance.)

We present our variable and clause gadgets and then show the correctness of the reduction. Let n be the number of variables in F, let m be the number of clauses in F, and, for $i \in [n]$, let p_i be the number of occurrences of the variable x_i in F.

Variable gadget. For every $i \in [n]$, we take as gadget for the variable x_i a copy H_i of the complete bipartite graph $K_{6p_i,3}$. Let A_i and B_i be the two sets in the bipartition of the vertex set of H_i such that $|A_i| = 6p_i$ and $|B_i| = 3$; see Figure 5. Recall that every bipartite graph admits a forest storyplan; see Example 4.

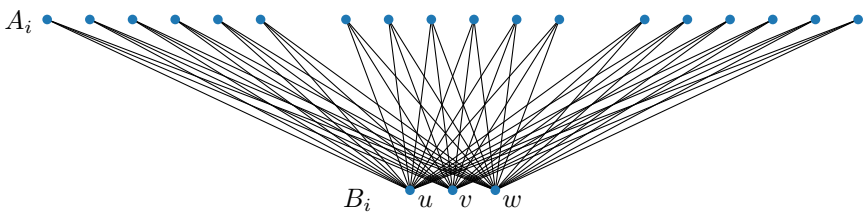


Figure 5 The gadget $H_i = K_{18,3}$ for a variable x_i that occurs three times in F.

We know that, for every $i \in [n]$, exactly one of A_i and B_i is completely visible in some frame of an outerplanar storyplan. This is due to Lemma 5 and to the fact that $6p_i > 3$. We call this set in the bipartition *fix* and the other set *flex*. No two vertices of the flex set are

visible in the same frame. For every $i \in [n]$, we assign the variable x_i to *true* if A_i is fix, and to *false* if A_i is flex. For each occurrence of x_i in a clause of F, we use six unique vertices of A_i in the corresponding clause gadget.

Clause gadget. We first construct a gadget for a clause that consists of two variables, then we construct a gadget for a clause that consists of three variables.

Let $C_t = (x_i, x_j)$ with $i, j \in [n]$, $i \neq j$, and $t \in [m]$ be a clause of F that consists of two (different) variables (assuming that such a clause exists). The gadget for C_t consists of six vertices of A_i and six vertices of A_j such that the graph induced by these twelve vertices is the complete bipartite graph $K_{6,6}$. Note that, by Lemma 5, exactly one of the sets in the partition of the vertex set of this $K_{6,6}$ has to be fix. If the fix part appears first, then the vertices from the flex part appear one by one, and we obtain a valid forest storyplan for such a clause gadget. Note that the fix part may stay visible after the last vertex of the flex part disappears. We use this observation later to show how to synchronize different gadgets.

Let $C_t = (x_i, x_j, x_k)$ with $i, j, k \in [n]$, $i \neq j \neq k \neq i$, and $t \in [m]$ be a clause of F that consists of three different variables. The gadget I_t for C_t is the graph depicted with black edges inside the yellow disk in Figure 6. The vertex set of the graph I_t is $A_i \cup A_j \cup A_k$; the three sets (of size 6) are shown in different colors in Figure 6. Note that every cycle of I_t contains at least one vertex of each color. The graph I_t admits a forest storyplan if exactly one of A_i, A_j, and A_k is fix. We order the vertices as shown in Figure 6: first, the vertices of the fix part appear (the blue vertices in Figure 6), then the remaining vertices appear one by one. Note that the fix part may stay visible after the last vertex of the flex parts disappears.

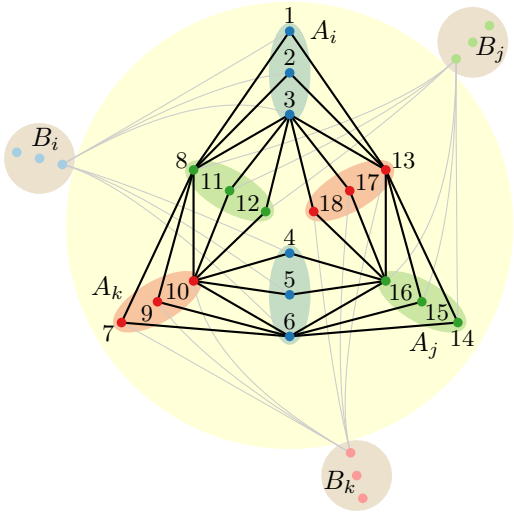


Figure 6 The gadget for the clause $x_i \vee x_j \vee x_k$ (inside the large yellow disk). The blue, green, and red vertices are the vertices of A_i, A_j, and A_k, respectively. The numbers correspond to an ordering of the vertices that yields a forest storyplan; in this case, A_i is fix, and A_j and A_k are flex. For clarity, not all edges (gray) to adjacent vertices (in B_i, B_j, and B_k) are shown.

Correctness. We first show that if every clause in F contains exactly one *true* variable, then a forest storyplan of G_F exists. In order to synchronize different clause gadgets, the vertices of G_F appear in three phases, in the following order: in the first phase, for each $i \in [n]$, the fix part of the variable gadget H_i appears. In the second phase, for each $t \in [m]$,

the flex parts of the variable gadgets involved in the clause C_t appear (and immediately disappear) in the order described in the paragraph "Clause gadget". In the third phase, for each $i \in [n]$ such that x_i is set to *true*, the vertices in B_i appear (and immediately disappear) one after the other (since they are flex). Note that this determines an ordering of the vertices of G_F.

We now assume that each clause contains exactly one *true* variable. Note that the vertices of the fix parts are not connected, thus in the first phase, we obtain a set of isolated vertices and hence a set of frames of a valid forest storyplan. After adding the flex parts in the second phase, we further obtain frames of a valid forest storyplan. To see this, let $t \in [m]$, and let h be a vertex of one of the flex parts of the clause gadget I_t. As mentioned in the paragraph "Clause gadget" above, the frame in which h (together with its neighbors from I_t) appears consists of a drawing of a forest. Apart from its neighbors in I_t, h is adjacent only to the three vertices of the fix part of the corresponding variable gadget, and when h is visible, these three vertices are only adjacent to h. Thus, the frame in which v appears consists of a drawing of a forest. Finally, each vertex u of a flex part ($u \in B_i$ for some $i \in [n]$) that is added in the third phase is adjacent to vertices of a fix part (A_i) that are only adjacent to u when u is visible. Note that, since we do not insist on straight-line edges, a partial drawing of a forest can always be extended without introducing crossings. Thus, the corresponding frame consists of a crossing-free drawing of a forest.

It remains to show that G_F has no outerplanar storyplan if in each variable assignment either zero, two, or three variables are assigned *true* in some clause C_t, $t \in [m]$. We argue that the corresponding clause gadget itself or, in the gadget of a variable x_i that occurs in C_t, the graph induced by B_i does not admit an outerplanar storyplan.

Recall that if a clause consists of only two variables, the corresponding clause gadget admits a forest storyplan only if exactly one part of the gadget is fix (by Lemma 5). So we now consider a clause $C_t = (x_i, x_j, x_k)$ with $t \in [m]$ and $i, j, k \in [n]$ that consists of three different variables. Recall that I_t is the clause gadget corresponding to C_t and that its vertices are contained in $A_i \cup A_j \cup A_k$. Let $a, b \in A_i \cap V(I_t)$, $c, d \in A_j \cap V(I_t)$, and $e, f \in A_k \cap V(I_t)$ be the vertices of degree 6 in I_t. Furthermore, let $a', a'', b', b'' \in A_i \cap V(I_t)$, $c', c'', d', d'' \in A_j \cap V(I_t)$, and $e', e'', f', f'' \in A_k \cap V(I_t)$ be the corresponding vertices of degree 2 in I_t; see Figure 7.

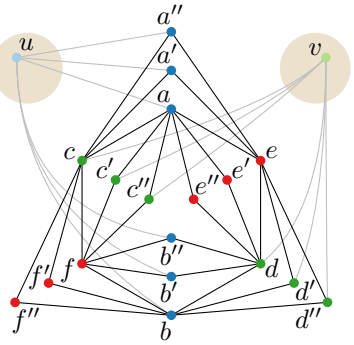


Figure 7 The clause gadget. The blue, green, and red vertices are the vertices of A_i, A_j, and A_k, respectively.

We distinguish the following three cases. In each case, we assume that G_F admits an outerplanar storyplan $\mathcal{S} = \langle \alpha, (D_r)_{r \in [n_F]} \rangle$ (where n_F is the number of vertices of G_F). We show that this leads to a contradiction.

Case 1. All parts A_i, A_j, and A_k are flex.

Recall that no two vertices of a flex part of the same variable gadget are visible at the same frame (due to Lemma 5). Without loss of generality, assume that a is the first degree-6 vertex of I_t that appears in $\mathcal{S}$, i.e., $\alpha(a) \leq \alpha(v)$ for every $v \in \{b, c, d, e, f\}$. Since a and b are from the same (flex) partition A_i, b can appear only after a disappears. In Figure 8, this precedence constraint is represented by arc ①. Thus, all neighbors of a (and, hence, c) appear before b (see arc ②). Vertex d can disappear only after b has appeared (see ③). Since c and d both belong to A_j, they cannot appear simultaneously, so d must appear after c has disappeared (see ④). We use a similar argument for e and f. Since e is a neighbor of a, e appears before b appears (see ⑤). Since f is a neighbor of b, f cannot disappear before b has appeared (see ⑥). Therefore, f appears after e has disappeared (see ⑦) because both belong to A_k. However, c cannot disappear before its neighbor f has appeared (see ⑧) and e cannot disappear before its neighbor d has appeared (see ⑨). Now the circular sequence ④, ⑨, ⑦, ⑧ of precedence constraints constitutes the desired contradiction.

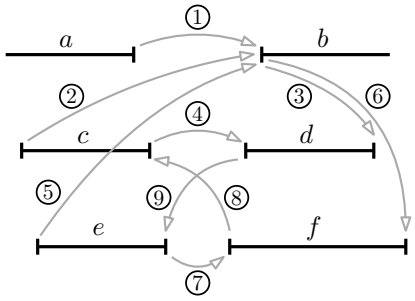


Figure 8 Precedence constraints between the start and end points of the visibility intervals of vertices of the clause gadget (in Case 1).

Case 2. All parts A_i, A_j, and A_k are fix.

As $K_{2,3}$ is not outerplanar, for each subgraph isomorphic to $K_{2,3}$, it holds that its five vertices do not appear in the same frame. On the other hand, as in each of the six occurrences of $K_{2,3}$ in I_t, the three vertices of degree 2 form always a subset of either A_i, A_j, or A_k which all are fix, such a $K_{2,3}$ can only be realized if its two vertices of degree 3 do not appear simultaneously. Therefore, a, d, and f have pairwise disjoint lifespans, as well as b, c, and e.

Assume without loss of generality that $\omega(a) < \alpha(d)$ (i.e., a disappears before d appears) and $\omega(d) < \alpha(f)$. Consequently, $\omega(b) < \alpha(c)$ and $\omega(c) < \alpha(e)$ because A_i and A_j are fix and, therefore, a and b appear in a common frame (analogously for $c, d \in A_j$ and $e, f \in A_k$).

Assume first that $\omega(a) \leq \omega(b)$. Then, since c is a neighbor of a, $\alpha(c) \leq \omega(a)$, and hence $\alpha(c) \leq \omega(b)$. This is a contradiction to $\omega(b) < \alpha(c)$ as observed above.

Now assume that $\omega(b) \leq \omega(a)$. Then, since d is a neighbor of b, $\alpha(d) \leq \omega(b)$, and hence $\alpha(d) \leq \omega(a)$. This is a contradiction to $\omega(a) < \alpha(d)$ as observed above.

Case 3. A_i, A_j are fix, and A_k is flex.

In order to show that G_F does not yield an outerplanar storyplan, consider the clause gadget I_t. Let I'_t be a subgraph of G_F induced by the vertices of I_t and two additional vertices, a vertex $u \in B_i$ and a vertex $v \in B_j$. Note that both partitions B_i and B_j are flex. Recall that all vertices of a fix part of a variable gadget are visible at some frame (due to Lemma 5) and vertices of each flex part have disjoint lifespans, in particular also vertices e and f.

Observe that in I'_t, $K_{3,3}$ appears as the following subgraphs with these vertex partitions:

(C1) $\{a, a', a''\}, \{u, c, e\}$
(C2) $\{b, b', b''\}, \{u, d, f\}$
(C3) $\{c, c', c''\}, \{v, a, f\}$
(C4) $\{d, d', d''\}, \{v, b, e\}$

Since A_i, A_j are fix, this implies that the vertices in each of the triples $\{u, c, e\}$, $\{u, d, f\}$, $\{v, a, f\}$, and $\{v, b, e\}$ have pairwise disjoint lifespans.

Assume first that u disappears before c appears, i.e. $\omega(u) < \alpha(c)$. The path u, a, c, v in I'_t yields $\alpha(a) \leq \omega(u) < \alpha(c) \leq \omega(a)$ and since the lifespans of a and v are disjoint, but v is adjacent to c, we get $\omega(a) < \alpha(v)$. Thus, the lifespan of u strictly precedes the lifespan of v.

Both e and f have disjoint lifespans from the lifespans of u and v, so they must appear after u disappears and both disappear before v appears as witnessed by the paths u, a, e, d, v and u, b, f, c, v in I'_t.

As e and f are in A_k, which is flex, their lifespans are disjoint. If $\omega(e) < \alpha(f)$, then since b is a common neighbor of u and f, the lifespans of b and e intersect, which contradicts their occurrence in (C4). Otherwise, if $\omega(f) < \alpha(e)$, then since a is a common neighbor of u and e, the lifespans of a and f intersect, which contradicts their occurrence in (C3). In both cases, we have derived a contradiction to the existence of an outerplanar storyplan.

Assume now that c disappears before u appears, i.e. $\omega(c) < \alpha(u)$. We will see that we can apply the symmetric argument by switching the roles of the vertices. The path v, c, a, u in I'_t yields $\alpha(c) \leq \omega(v) < \alpha(a) \leq \omega(c)$ and since the lifespans of c and u are disjoint, but u is adjacent to a, we get $\omega(c) < \alpha(u)$. Thus, the lifespan of v strictly precedes the lifespan of u.

Both e and f have disjoint lifespans from the lifespans of u and v. By the same argument as above, e and f appear between v and u. With a symmetric argument, we end up with a similar contradiction as above.

We have shown that the graph G_F admits a forest storyplan if the formula F is satisfiable. Otherwise G_F does not admit even an outerplanar storyplan. Observe that the size of G_F is linear in the size of F: there is a constant number of vertices and edges in each clause gadget, the total number of vertices in sets $B_1, \ldots, B_n$ is $3n$, and the total number of occurrences of all variables is at most $3m$, which bounds the number of edges outside the clause gadgets by $18m$.

As a result we obtain the NP-hardness of both problems.

▶ **Theorem 9.** *Both* Forest Storyplan *and* Outerplanar Storyplan *are* NP*-hard.*

In the following, we show that both problems are in NP. The proof is based on the corresponding proof for Planar Storyplan of Binucci et al. [6].

▶ **Theorem 10.** *Both* Forest Storyplan *and* Outerplanar Storyplan *are in* NP*.*

Proof. In order to show that Forest Storyplan is in NP, it is enough to guess the vertex order and check in each step whether the graph is a forest. We do not need to check the drawings in each step since we can always extend a drawing of a tree by adding new vertices and edges avoiding crossings.

For Outerplanar Storyplan we need to additionally check the drawings in each step and their "consistency" – the common parts of the drawings in each pair of consecutive steps have to coincide. To this end, we rely on the proof of [6, Theorem 3], where Binucci et al. showed that Planar Storyplan is in NP, but instead of looking for planar embeddings, we look for outerplanar ones. ◀

5 Parameterized algorithms

In this section, we observe that the existing parametrized algorithms (with respect to two common structural parameters, namely vertex cover number and feedback edge set number) for Planar Storyplan [6] can be modified to obtain corresponding algorithms for Forest Storyplan and for Outerplanar Storyplan.

5.1 FPT algorithm with respect to the vertex cover number

First we show how to obtain FPT algorithms with respect to the vertex cover number for Forest Storyplan and Outerplanar Storyplan using the corresponding FPT algorithm for Planar Storyplan of Binucci et al. [6, Theorem 4]. A subset C of the vertex set of a graph G is a *vertex cover* of G if every edge of G is incident to a vertex in C, and the *vertex cover number* of G is the minimum size of a vertex cover of G. We use the same notation as Binucci et al.

Without loss of generality, we assume that the input graph G does not contain isolated vertices, as such vertices do not affect the existence of a storyplan (of any kind). Let C be a vertex cover of size k of G. For $U \subseteq C$, a vertex $v \in V \setminus C$ is of *type* U if $N(v) = U$, where $N(v)$ denotes the set of neighbors of v in G. This defines an equivalence relation on $V \setminus C$ and in particular partitions $V \setminus C$ into at most $\sum_{i=1}^{k} \binom{k}{i} = 2^k - 1 < 2^k$ distinct types. Let V_U denote the set of vertices of type U.

In order to obtain a kernel of G, Binucci et al. [6] used the property of planar storyplans for bipartite graphs (see Lemma 3) to define three reduction rules.

(R1) *If there exists a type U such that $|U| = 1$, then pick an arbitrary vertex $x \in V_U$ and remove it from G.*

(R2) *If there exists a type U such that $|U| = 2$ and $|V_U| > 1$, then pick an arbitrary vertex $x \in V_U$ and remove it from G.*

(R3) *If there exists a type U such that $|U| \geq 3$ and $|V_U| > 3$, then pick an arbitrary vertex $x \in V_U$ and remove it from G.*

We modify the rules according to analogous properties of outerplanar and forest storyplans (see Lemma 5). To this end, we keep the rule R1 for both Forest Storyplan and Outerplanar Storyplan, and we combine rules R2 and R3 in a slightly different manner for Forest Storyplan (rule R2$^{\mathsf{f}}$) and for Outerplanar Storyplan (rule R2°). We will see that, for the more restricted structures of forests and outerplanar graphs, it suffices to have one rule (R2$^{\mathsf{f}}$ or R2°) instead of two rules (R2 and R3).

(R2$^{\mathsf{f}}$) *If there exists a type U such that $|U| \geq 2$ and $|V_U| > 2$, then pick an arbitrary vertex $x \in V_U$ and remove it from G.*

(R2°) *If there exists a type U such that $|U| \geq 2$ and $|V_U| > 3$, then pick an arbitrary vertex $x \in V_U$ and remove it from G.*

Analogously to [6, Lemma 7], we next show that a graph obtained from G by applying one of the reduction rules R1 and R2$^{\mathsf{f}}$ (R2°) admits a forest (outerplanar) storyplan if and only if G does.

▶ **Lemma 11.** *Let G' be a graph obtained from G by applying one of the reduction rules R1 and R2$^{\mathsf{f}}$ (R2°). Then, G' admits a forest (outerplanar) storyplan if and only if G does.*

Proof. Clearly, G' is a subgraph of G. Hence, if G admits a forest (outerplanar) storyplan, then also G' does due to Property 2. It remains to show that if G' admits a forest (outerplanar) storyplan, so does G.

Let x be the only vertex of $V(G) \setminus V(G')$. If R1 has been applied to obtain G' from G, then the argument is the same as in [6, Lemma 7]: Consider the only neighbor u of x in G. In the storyplan $\mathcal{S}'$ of G', u appears in some step i_u. Obtain a forest (outerplanar) storyplan $\mathcal{S}$ of G from $\mathcal{S}'$ as follows. Let x appear in step $i_u + 1$ and let it disappear right after. Draw x sufficiently close to u such that the straight-line segment ux does not interfere with the rest of the drawing. The rest of the storyplan remains unchanged (just the indices of the time steps after i_u increase by 1).

Next, consider the case that R2$^\mathsf{f}$ has been applied. Let $U = \{u_1, u_2, \dots\} \subseteq C$ be the neighbors of x, and let $y, z \in V(G') \setminus C$ be two "twins" of x being of the same type V_U as x. By Lemma 5, either the vertices in U, or y and z are simultaneously visible in a forest storyplan $\mathcal{S}'$ of G'. Suppose the vertices in U are all visible in the steps $[i, j]$. Let y and z be visible in the steps $[i_y, j_y]$ and $[i_z, j_z]$, respectively. W.l.o.g., $i_y \leq j_y < i_z \leq j_z$. Obtain a forest storyplan $\mathcal{S}$ of G from $\mathcal{S}'$ as follows. Let x appear in step $j_y + 1$ and draw x at the position of y and let all edges to the neighbors in U coincide with those of y. Then, x disappears in the next step. The rest of the storyplan remains the same (but the indices increase by 1). If $\mathcal{S}'$ was a forest storyplan, so is $\mathcal{S}$. Now suppose that y, z are both visible in the steps $[i, j]$. Clearly, at most one vertex of U is visible at a time (see Lemma 5). For each step $s \leq i$, the new storyplan $\mathcal{S}$ is the same as $\mathcal{S}'$. For each step $s > i$, the new storyplan $\mathcal{S}$ is the same as $\mathcal{S}'$ in step $s - 1$ but with x being additionally visible in the steps $[i + 1, j + 1]$. In these steps, x is located next to y. In each step, there is at most one edge from y to a vertex $u \in U$. Add another edge from x to u that follows the curve of the edge yu sufficiently close such that no intersections arise. This shows the correctness of the rule R2$^\mathsf{f}$.

We apply the analogous arguments to the rule R2°. The only difference is that we have besides y and z a third vertex z'. If the vertices in U are all visible simultaneously, then again draw x at the position of y. If y, z, and z' are visible simultaneously, then again draw x (and its incident edges) next to the location of y (and its incident edges). If $\mathcal{S}'$ was an outerplanar storyplan, so is the resulting storyplan $\mathcal{S}$. ◀

Using Lemma 11, we now prove the analogous result to [6, Theorem 4]:

▶ **Theorem 12.** *Deciding whether a graph G admits a forest (or outerplanar) storyplan, and computing such a storyplan (if any) can be done in $2^{2^{\mathcal{O}(k)}} + \mathcal{O}(n^2)$ time, where n is the number of vertices of G and k is the vertex cover number of G.*

Proof. Compute a vertex cover C of size k in G in $\mathcal{O}(2^k + kn)$ time [10]. Then, group the vertices in $V(G) \setminus C$ by type into at most n sets. Apply the rules R1 and R2$^\mathsf{f}$ (R2°) exhaustively. This can be one in $\mathcal{O}(n)$ time per application. Each application reduces the number of vertices by 1, so there are $\mathcal{O}(n)$ applications and a total running time of $\mathcal{O}(n^2)$ for this step. We obtain a kernel $G^\star$ of size $\mathcal{O}(2^k)$. In $G^\star$, compute a forest (outerplanar) storyplan by brute force: Guess a total order of the vertices; this defines the lifespan of every vertex and a subgraph for every frame. For each step $i = 2, 3 \dots, |V(G^\star)|$, maintain the set of all forest (outerplanar) embeddings by taking the ones from step $i - 1$ and adding the vertex v that appears in step i at all possible positions in the rotation systems of v's neighbors. Discard the ones not being forest (outerplanar) embeddings and remove duplicates. There are $|V(G^\star)|! \in (2^k)^{\mathcal{O}(k2^k)} \subseteq 2^{2^{\mathcal{O}(k)}}$ total orders of the vertices. The number of totals orders of the vertices also bounds the number of rotation systems. Hence, this step takes $2^{2^{\mathcal{O}(k)}} \cdot 2^{2^{\mathcal{O}(k)}} = 2^{2^{\mathcal{O}(k)}}$ time in total. If there is at least one forest (outerplanar) storyplan

after step $|V(G^\star)|$, there is also such a storyplan for G due to Lemma 11. Note that the proof of Lemma 11 is constructive and a storyplan of G can be obtained from that of $G^\star$ in $O(n^2)$ time. ◀

5.2 FPT algorithm with respect to the feedback edge set number

Binucci et al. [6, Theorem 5] also derived an FPT algorithm with respect to the feedback edge set number. A subset F of the edge set of a graph G is a *feedback edge set* if $G - F$ is acyclic, and the *feedback edge set number* of G is the minimum size of a feedback edge set of G. In their algorithm, Binucci et al. defined the following reduction rules to obtain a kernel of a graph G of size $\mathcal{O}(f)$, where f is the feedback edge set number of G.

(RA) *If there exists a vertex of degree one, then remove it from G.*
(RB) *If, for $k \geq 3$, there exists a path with $k+2$ vertices such that its k inner vertices all have degree two, then remove its inner vertices from G.*

Their algorithm also works for Forest Storyplan and Outerplanar Storyplan: After exhaustively applying both rules, the resulting graph has size $O(f)$. This means that we can guess a forest or outerplanar storyplan if it exists. Into this storyplan, we re-insert the vertices that have been removed by applying the rules as follows. The next lemma corresponds to [6, Lemma 8].

▶ **Lemma 13.** *Let G' be a graph obtained from G by applying one of the reduction rules RA and RB. Then, G' admits a forest (or outerplanar) storyplan if and only if G does.*

Proof. Clearly, G' is a subgraph of G. Hence, if G admits a forest (outerplanar) storyplan, then also G' does due to Property 2. It remains to show that if G' admits a forest (outerplanar) storyplan, so does G.

If RA has been applied to obtain G' from G, then the argument is the same as in Lemma 11: Let x be the only vertex of $V(G) \setminus V(G')$ and let u be its only neighbor. Obtain a forest (outerplanar) storyplan of G from such a storyplan of G' as follows. Let x appear right after u appears and let x disappear in the next step. Draw x sufficiently close to u such that the straight-line segment ux does not interfere with the rest of the drawing. The rest of the storyplan remains unchanged.

Now suppose that RB has been applied to obtain G' from G and let $\langle r, u_1, \ldots, u_k, t\rangle$ be the path in G such that $r, t \in V(G')$ and $u_1, \ldots, u_k \in V(G) \setminus V(G')$. Let $\mathcal{S}'$ be a forest (or outerplanar) storyplan of G'. We next describe how to obtain a forest (or outerplanar) storyplan $\mathcal{S}$ of G. Let $[i_r, j_r]$ and $[i_t, j_t]$ be the lifespans of r and t, respectively, in $\mathcal{S}'$. W.l.o.g., $i_r < i_t$. Let u_1 appear in step $i_r + 1$ and let u_k appear in step $i_t + 2$. Draw u_1 close to r and u_k close to t such that the straight-line segments ru_1 and u_kt do not interfere with the rest of the drawing. For each step $s \leq i_r$, the new storyplan $\mathcal{S}$ is the same as $\mathcal{S}'$. For each step $s \in [i_r + 1, i_t + 1]$, $\mathcal{S}$ is the same as $\mathcal{S}'$ in step $s - 1$ with u_1 being additionally visible. For each step $s > i_t + 1$, $\mathcal{S}$ is the same as $\mathcal{S}'$ in step $s - 2$ with u_1 and u_k being additionally visible. Finally, we adjust the end of $\mathcal{S}$ by letting the remaining vertices appear in order: for each $i \in [2, k-1]$, let u_i appear in step $n' + 1 + i$ where $n' = |V(G')|$. Note that this is still a forest (or outerplanar) storyplan since after step $n' + 2$, only u_1 and u_k are still visible together with the vertices of the path $\langle u_2, \ldots, u_{k-1}\rangle$. It is easy to arrange $u_2, \ldots, u_k$ along the line segment between u_1 and u_k in a non-crossing way. ◀

This yields a linear-size kernel and leads to the following theorem, which directly corresponds to a result by Binucci et al. [6, Lemma 9 and Theorem 5].

▶ **Theorem 14.** *Deciding whether a graph G admits a forest (or outerplanar) storyplan, and computing such a storyplan (if any) can be done in $2^{\mathcal{O}(f \log f)} + \mathcal{O}(n^2)$ time, where n is the number of vertices of G and f is the feedback edge set number of G.*

Proof. Compute a minimum-size feedback edge set F of G in linear time. Note that this is possible by taking the complement of a (largest) spanning forest, which is a linear-time problem. Apply the rules RA and RB exhaustively. This can be one in $\mathcal{O}(n)$ time per application. Each application reduces the number of vertices by at least 1, so there are $\mathcal{O}(n)$ applications and a total running time of $\mathcal{O}(n^2)$ for this step. Let the resulting graph be $G^\star$. Let $F^\star \subseteq F$ be the set of edges of F in $G^\star$. Clearly, removing $F^\star$ from $G^\star$ yields a forest $H^\star$. Observe that $H^\star$ has at most $2f$ leaves because any leaf in $H^\star$ is an endpoint of an edge in $F^\star$ as otherwise it would not be in $G^\star$ due to RA. Moreover, in any forest, the number of vertices with degree at least 3 is at most the number of leaves minus two. Hence, the number of degree-1 and degree-at-least-3 vertices in $H^\star$ is at most $4f-2$. Between those vertices, there are at most two neighboring vertices of degree 2 due to RB, that is, at most $2(4f-3)$ degree-2 vertices. Hence, $|E(H^\star)| < |V(H^\star)| \leq 12f-8$ and $|E(G^\star)| < 13f-8$. So, $G^\star$ is a kernel of size $\mathcal{O}(f)$. As in Theorem 12, compute a forest (outerplanar) storyplan of $G^\star$ by brute force: Guess a total order of the vertices; this defines the lifespan of every vertex and a subgraph for every frame. For each step $i = 2, 3 \ldots, |V(G^\star)|$, maintain the set of all forest (outerplanar) embeddings by taking the ones from step $i-1$ and adding the vertex v that appears in step i at all possible positions in the rotation systems of v's neighbors. Discard the ones not being forest (outerplanar) embeddings and remove duplicates. There are $|V(G^\star)|! \in 2^{\mathcal{O}(f \log f)}$ total orders of the vertices. The number of totals orders of the vertices also bounds the number of rotation systems. Hence, this step takes $2^{\mathcal{O}(f \log f)} \cdot 2^{\mathcal{O}(f \log f)} = 2^{\mathcal{O}(f \log f)}$ time in total. If there is at least one forest (outerplanar) storyplan after step $|V(G^\star)|$, there is also such a storyplan for G due to Lemma 13. Note that the proof of Lemma 13 is constructive and a storyplan of G can be obtained from that of $G^\star$ in $O(n^2)$ time. ◀

6 Outerplanar storyplans

In this section, we present families of graphs that admit outerplanar storyplans.

▶ **Theorem 15.** *Every partial 2-tree admits a geometric outerplanar storyplan, and such a storyplan can be computed in linear time.*

Proof. Due to Property 2, it suffices to prove the statement for 2-trees.

Let G be a 2-tree. Hence, there exists a *stacking order* $\sigma = \langle v_1, \ldots, v_n \rangle$ of the vertex set of G. In other words, G can be constructed as follows: we start with v_1, v_2, v_3 forming a K_3 and then, for $i \geq 4$, v_i is *stacked* on an edge $v_k v_\ell$ with $k, \ell < i$, that is, v_i is connected to v_k and v_ℓ by edges. We claim that we can choose a vertex order σ' and an embedding $\mathcal{E}$ of G such that σ' (together with $\mathcal{E}$) defines an outerplanar storyplan. Moreover, we can obtain a straight-line drawing of G with embedding $\mathcal{E}$ in linear time [13, 22]. Let Γ be such a drawing. For the outerplanar storyplan that we construct we use the positions of vertices and edges as in Γ. This yields a geometric storyplan. Figure 9(a) shows a 2-tree with a stacking order (that is not an outerplanar storyplan).

To show that an outerplanar storyplan always exists, consider the *stacking tree* $T_{G,\sigma}$ of G. The root of $T_{G,\sigma}$ represents the triangle $\Delta v_1 v_2 v_3$ given by the first three vertices of σ. For $i \in \{4, 5, \ldots, n\}$, let $v_k v_\ell$ with $k < \ell < i$ be the edge on which v_i was stacked. Then the node of $T_{G,\sigma}$ that represents v_i is a child of the node that represents v_ℓ. Note that if $\ell \leq 3$, then v_i is a child of the root. Figure 9(b) shows the stacking tree of the 2-tree in Figure 9(a).

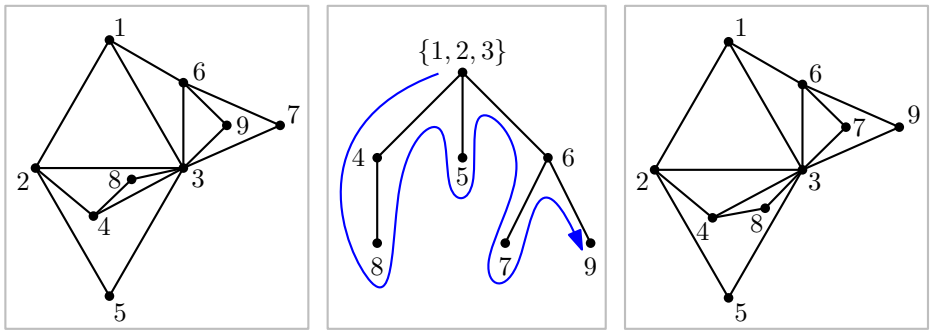
From $T_{G,\sigma}$, we obtain a vertex order $\sigma' = \langle v'_1, v'_2, \ldots, v'_n \rangle$ that is an outerplanar storyplan as follows; see Figure 9(c). Let $v'_1 = v_1$, $v'_2 = v_2$, and $v'_3 = v_3$. Now, we traverse $T_{G,\sigma}$ in depth-first order and, for each node τ of $T_{G,\sigma}$, we add to σ' the vertex of G that τ represents. We claim that for σ', we can choose an embedding $\mathcal{E}$ (defined implicitly next) of G such that every frame is outerplanar. Note that the first three vertices form a triangle, which always admits an outerplanar drawing. Now consider v'_i for $i \in \{4, 5, \ldots, n\}$. Our invariant is that, before the i-th frame starts, the parent p of v'_i in $T_{G,\sigma}$ has degree 2 in the current outerplanar drawing and lies on the outer face. This implies that v'_i can be added to the outer face because it is stacked onto an edge of the outer face resulting again in an outerplanar drawing. Of course, for $i = 4$, our invariant is satisfied. If $p = v'_{i-1}$, then our invariant is trivially satisfied. Otherwise, let $p = v'_j$ for some $j < i - 1$. Observe that, for $k \in \{j + 1, \ldots, i - 1\}$, each v'_k will have disappeared by the end of the $(i - 1)$-th frame. This is due to the fact that v'_k is not an ancestor of v_i, which means that all of the neighbors of v'_k have already been introduced to the storyplan due to the depth-first pre-order traversal. Essentially, every frame given by σ' shows a subpath of $T_{G,\sigma}$, which is a sequence of stacked triangles admitting an outerplanar drawing. ◀

▶ **Theorem 16.** *Every subcubic graph except K_4 admits a geometric outerplanar storyplan with at most five edges in each frame, and such a storyplan can be computed in linear time.*

Proof. Due to Property 2, it suffices to prove the statement for cubic graphs.

We can assume that the given cubic graph G (which is not K_4) is connected. Otherwise, we consider each connected component separately. To construct an outerplanar storyplan, below, we will order the vertices $v_1, \ldots, v_n$ of G such that the resulting sequence of graphs $\langle G_1, G'_1 \ldots, G_{n-1}, G'_{n-1}, G_n \rangle$ has the following property: for $4 \leq i \leq n - 1$, G'_i contains at most two edges. Only G'_3 may be a triangle and would thus contain three edges. Then we show how to obtain outerplanar drawings $D_1, D'_1 \ldots, D_{n-1}, D'_{n-1}, D_n$ of the graphs $G_1, G'_1 \ldots, G_{n-1}, G'_{n-1}, G_n$, respectively. For $i \in [n]$, let $H_i = G[\{v_1, \ldots, v_i\}]$.

We pick the first vertex v_1 arbitrarily. For $2 \leq i \leq n$, consider the vertices of G'_{i-1} that have maximum degree in H_{i-1}. Among those, let v be a vertex that additionally has the largest degree in G'_{i-1}. Then, let v_i be a neighbor of v in $G - \{v_1, \ldots, v_{i-1}\}$. Note that v always has such a neighbor, otherwise v would already be completed and, hence, would not be in G'_{i-1}. The intuition behind this choice is that we want to remove v from the drawing as soon as possible.



(a) (b) (c)

Figure 9 (a) Straight-line drawing of a 2-tree G with stacking order $\sigma = \langle 1, 2, 3, 4, 8, 5, 6, 7, 9 \rangle$; (b) the stacking tree $T_{G,\sigma}$ of G and σ yields a vertex order σ'; (c) an embedding of G that together with σ' defines an outerplanar storyplan.

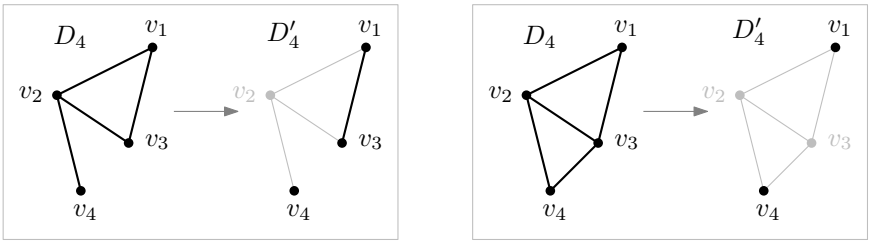


Figure 10 Illustration of the special case that G'_3 contains three edges. Then v_4 has one neighbor in H_3 (left subfigure) or two neighbors in H_3 (right subfigure), but G'_4 has at most one edge. Gray vertices and edges are not part of the current graphs, but were visible in the previous step.

Note that, for every $i \in [n]$, the graph H_i is connected. This is due to the fact that G is connected and the new vertex v_i is always adjacent to a vertex in G'_{i-1}.

We claim that, for $4 \leq i \leq n-1$, the graph G'_i contains at most two edges and, if G'_i contains exactly two edges, then both edges are incident to v_i. If our claim holds, then, for $i \in [n]$, G_i contains at most five edges. Indeed, even if G'_3 has three edges (that is, G'_3 is a triangle; see Figure 10), then G_4 still has at most five edges since G is not K_4.

We show our claim by induction over i.

For the base of the induction, observe that G'_1 $(= G_1)$ contains only vertex v_1 and no edges and G'_2 $(= G_2)$ is the edge v_1v_2. If G'_3 $(= G_3)$ is a triangle, then G'_4 has at most one edge (see Figure 10). Otherwise G'_3 is the path $\langle v_1, v_2, v_3\rangle$, G_4 is a 3-star with apex v_2, and G'_4 consists of the three independent leaves of G_4.

Now, for the inductive step, consider $i \geq 4$. There are three cases depending on the degree of v in G'_{i-1}; see Figure 11.

(C1) Vertex v does not have any neighbors in G'_{i-1}.
By the choice of v, this implies that there are no edges in G'_{i-1}. Hence, all edges in G_i are new and incident to v_i. If v_i has three neighbors in G_i, then v_i will disappear, and there are no more edges in G'_i. Otherwise, G'_i has at most two edges, which are both incident to v_i.

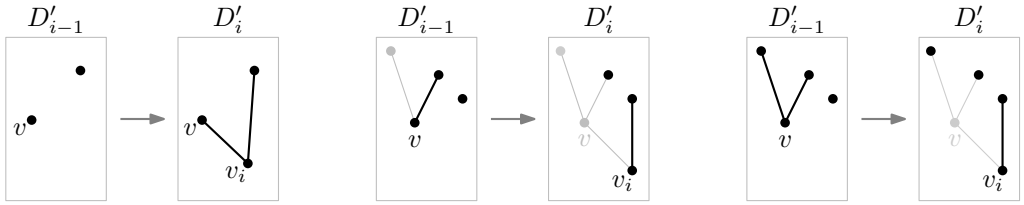
(C2) Vertex v has one neighbor in G'_{i-1}.
If v has degree 2 in H_{i-1}, then v disappears in the next step and G'_i does not contain it. Since v_i has at most one edge that stays in G'_i, the number of edges in G'_i is not larger than in G'_{i-1}. If v has degree 1 in H_{i-1}, then, by the choice of v, all other vertices in G_{i-1} have degree at most 1 in H_{i-1}. Hence, $i = 3$, that is, v and its neighbor are the first two vertices that we introduced.

(C3) Vertex v has two neighbors in G'_{i-1}.
In this case, the two edges incident to v are the only edges in G'_{i-1}. Then v disappears as v_i is its last neighbor. Therefore, G'_i contains at most one edge that v_i may have introduced.

We have shown that, in each case, the number of visible edges in G'_i, for $4 \leq i \leq n-1$, is at most two. Note that, if there are two edges, then they share an endpoint.

It is obvious that the drawings D_1, D'_1, D_2, D'_2, D_3, and D'_3 are outerplanar even if they are straight-line. Now we want to show, for $i \in \{4, \ldots, n\}$, how to place v_i such that D_i is outerplanar and straight-line. We assume that v_i is connected to three visible vertices $u, v, w \in V(G'_{i-1})$. The other cases are easier and are covered by this case.

We know that there are at most two edges in G'_{i-1}. If there are edges, we may assume that they are connected to v. We consider the case where there are exactly two edges (v, v')



(a) case (C1) **(b)** case (C2) **(c)** case (C3)

Figure 11 Cases considered in the proof of Theorem 16. In each of them, we depict a situation where the number of edges in G'_i is maximized. Gray vertices and edges were visible in some previous steps.

and (v, v'') in G'_{i-1}; the other cases are covered by this one. Note that possibly u is v' and/or w is v''. In this case, the triangle $\triangle uvv_i$ and/or the triangle $\triangle vwv_i$ would appear in D_i. In any case, we place v_i in the vicinity of v such that none of the new edges intersects the visible ones and no visible vertex lies in one of the triangles $\triangle uvv_i$ and $\triangle vwv_i$ that we have potentially created. It is easy to see that such a placement exists.

Now assume that u, w, v', and v'' are pairwise different vertices. In this case G_i is a tree, and we need to avoid only edge intersections. Let $\ell_{v'}$ and $\ell_{v''}$ be rays that start in v and go through v' and v'', respectively. There are three cases: $\triangle vuw$ intersects neither $\ell_{v'}$ nor $\ell_{v''}$, it intersects both, or it intersects exactly one of them.

In the first two cases, we place v_i in the intersection I of the open halfplanes bounded by the lines through (v, u) and (v, w) that contain neither v' nor v''; see the gray regions in Figures 12a and 12b. Now assume that $\triangle vuw$ intersects one of the rays, say, $\ell_{v'}$. Let ℓ_u be the line that goes through u and v', and let ℓ_w be the line that goes through w and v'. Let h_u and h_w be the halfplanes bounded by ℓ_u and ℓ_w, respectively, that do not contain v. Let $h_{v''}$ be the halfplane bounded by $\ell_{v''}$ that contains v'. Let $I = h_u \cap h_w \cap h_{v''}$. Note that $h_u \cap h_w$ is a non-empty wedge with apex v'. Since $\ell_{v''}$ does not go through v' but $h_{v''}$ contains v', h is either a wedge or a (possibly unbounded) polygon with three sides. Hence, the interior I^- of $I \setminus \ell_{v'}$ is non-empty, and we place v_i in I^-; see Figure 12c, where (u, w) does not intersect (v, v'), and Figure 12d, where (u, w) intersects (v, v'). None of the new edges incident to v_i contains a vertex visible in D_i. Hence, we have found a position for v_i such that the drawing D_i is straight-line and outerplanar.

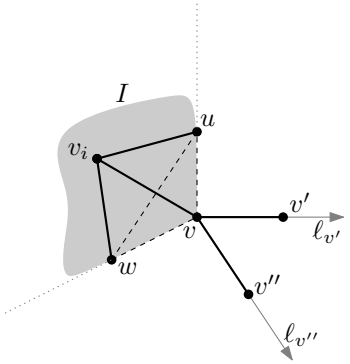
To argue the linear runtime, note that, for every $i \in [n]$, G_i has constant size. Therefore, we can pick v_i in constant time. Since the degree of v_i is at most 3, we can update H_i in constant time by using a standard graph data structure. ◀

7 Forest storyplans

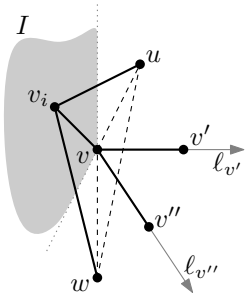
Clearly, any triangle is an obstruction for a graph to admit a forest storyplan. Interestingly, for planar and subcubic graphs this is the only obstruction for the existence of a forest storyplan as we show now.

▶ **Theorem 17.** *Every $\triangle$-free subcubic graph admits a geometric forest storyplan. Such a storyplan can be computed in linear time and has at most five edges per frame.*

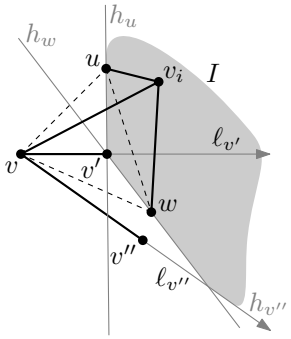
Proof. Due to Property 2, it suffices to prove the statement for $\triangle$-free cubic graphs. Recall the proof of Theorem 16. In that proof, we showed that, for $4 \leq i \leq n-1$, there are at most two edges in G'_i, and two edges may appear only if they share a vertex. In such a case, we always pick a shared vertex as a neighbor of the next vertex. Since our graph is $\triangle$-free,



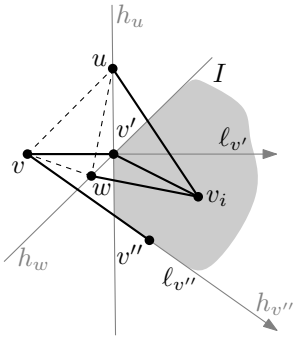
(a) $\triangle vuw$ intersects neither $\ell_{v'}$ nor $\ell_{v''}$.



(b) $\triangle vuw$ intersects both $\ell_{v'}$ and $\ell_{v''}$.



(c) $\triangle vuw$ intersects only $\ell_{v'}$, uw does not intersect (v, v').



(d) $\triangle vuw$ intersects only $\ell_{v'}$, (u, w) intersects (v, v').

Figure 12 Various cases for the placement of v_i. In each case, v_i can be placed in the gray region I to obtain a straight-line outerplanar drawing D_i.

adding a vertex will never make a cycle visible. Moreover, G'_3 cannot be a triangle and, thus, for every $i \in [n-1]$, G'_i contains at most two edges. ◀

As a warm-up for our main result regarding $\triangle$-free planar graphs (Theorem 19), we first present an algorithm for constructing storyplans for $\triangle$-free *outer*planar graphs.

▶ **Proposition 18.** *Every $\triangle$-free outerplanar graph admits a geometric forest storyplan, and such a storyplan can be computed in linear time.*

Proof. Let G be a $\triangle$-free outerplanar graph, and let Γ be an outerplanar straight-line drawing of G. Let $\sigma = \langle v_1, v_2, \ldots, v_n \rangle$ be the circular order of the vertices along the outer face of Γ (which can easily be determined in linear time [17, 18]). We claim that σ yields a forest storyplan of G. (The resulting storyplan will also be geometric because every frame is a subdrawing of Γ.)

To this end, we show that there is no frame where a complete face of Γ is visible. If this is true, then no frame contains a complete cycle. This is due to the fact that, in outerplanar graphs, the vertex set of every cycle contains the vertex set of at least one face. Let $F = \langle v_{i_1}, v_{i_2}, \ldots, v_{i_k} \rangle$ with $i_1 < i_2 < \cdots < i_k$ be a face of G. Since G is $\triangle$-free, we have $k \geq 4$. Note that v_{i_1} and v_{i_3} as well as v_{i_2} and v_{i_4} are not adjacent. Since G is outerplanar, v_{i_2} may be adjacent only to vertices that appear in σ between (and including) v_{i_1} and v_{i_3}. Therefore, v_{i_2} disappears before v_{i_4} appears. Hence it is indeed not possible that all vertices of the same face appear in a frame. ◀

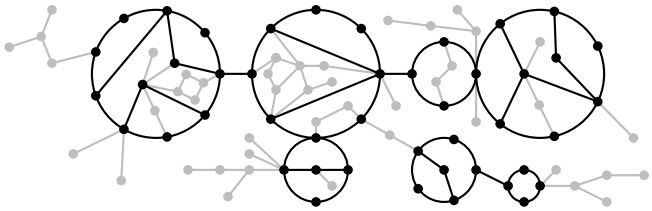


Figure 13 From an embedded $\triangle$-free planar graph G_i (black & gray), we obtain G_i' (black). Note that G_i' decomposes into seven simple cycles and two connected components. The outer edges and vertices of these connected components form cactus graphs.

Now we generalize the simple result above from $\triangle$-free outerplanar graphs to $\triangle$-free planar graphs. Note, however, that we do not guarantee linear running time any more.

▶ **Theorem 19.** *Every $\triangle$-free planar graph admits a geometric forest storyplan, and such a storyplan can be computed in quadratic time.*

Proof. Let G be a $\triangle$-free planar graph, and let Γ be a planar straight-line drawing of G. In the desired forest storyplan for G, we use the positions of the vertices in Γ.

We first give a rough outline of our iterative algorithm and then describe the details. In each iteration (which spans one or more steps of the storyplan that we construct), we *pick* a vertex on the current outer face (we explain below what the current outer face is). We add the picked vertex and its neighbors (if they are not visible yet) to the storyplan one by one. In this way, after each iteration, at least one vertex disappears, namely the one we picked.

For the ease of notation, we slightly adjust the definition of G_i in this proof. Namely, we let $G_1 = G$ and, for $i \in \{1, 2, \dots\}$, we let v_i be the vertex that we pick in iteration i. Then, we define G_{i+1} as the subgraph of G_i that we obtain after removing the vertices (and the edges incident to them) that disappear in iteration i; see Figure 14b. The algorithm terminates as soon as G_i is a forest and adds the remaining vertices in arbitrary order to the storyplan under construction. When we speak about the (current) outer face in step i, we mean the outer face of G_i in the drawing derived from Γ. We call vertices and edges incident to the (current) outer face *outer*. The others are *inner*.

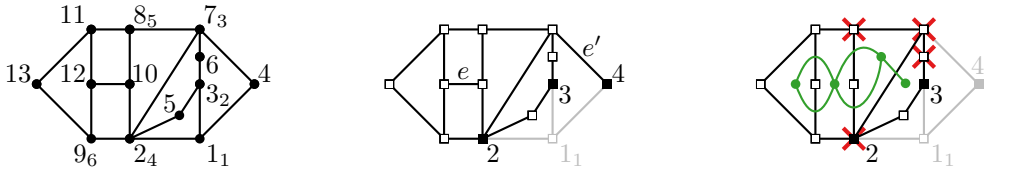
We always pick outer vertices. For this reason, only two types of vertices can potentially create cycles: the endpoints of *chords* (i.e., inner edges incident to two outer vertices) and the endpoints of *half-chords* (i.e., length-2 paths that connect two outer vertices via an inner vertex).

Let G_i' be the (embedded) subgraph of G_i (embedded according to Γ) that consists of all vertices and edges that lie on a simple cycle that bounds the outer face of G_i, plus every edge that connects two cycles, plus all chords and half-chords (and, thus, plus the inner vertices that lie on the half-chords) of G_i; see Figures 13 and 14c. For example, the edges e and e' of G_2 in Figure 14b are not part of G_2'. We say that a vertex of G_i' is *free* if it lies on the outer face and is not part of a chord or a half-chord.

Let H_i be the weak dual of G_i' (see Figure 14c), i.e., the (embedded) multigraph that has a vertex for each inner face of G_i' and an edge for each pair of inner faces that are incident to a common edge of G_i'. Note that H_i is outerplane and even has all edges on the outer face (since the inner vertices of G_i' form an independent set) and that H_i has no loops (since G_i' does not have leaves). We maintain the following invariants:

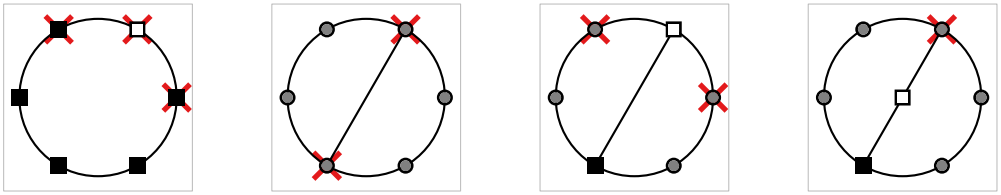
(I1) At no point in time, the set of visible edges on the outer face forms a cycle.

(I2) *During* iteration i, the only inner vertices that may be visible are those that are adjacent to v_i and to no other visible vertex on the outer face.



(a) △-free planar graph G with a forest storyplan **(b)** iteration 2: subgraph G_2 (black) of $G_1 = G$ **(c)** subgraph G_2' of G_2 with weak dual H_2 (green)

Figure 14 A △-free planar graph G where (a) shows a forest storyplan computed by our algorithm, (b) shows the result of the first iteration of the algorithm, and (c) shows the auxiliary graph for the second iteration. Subscripts refer to the iteration in which a vertex is picked. Red crosses mark vertices that may not be picked.



(a) rule 1 **(b)** rule 2 **(c)** rule 3 **(d)** rule 4

Figure 15 Rules that determine which vertices may not be picked (marked by red crosses). Black squares represent visible vertices, white squares represent invisible vertices, and gray disks represent vertices that may be visible or invisible.

(I3) *During* iteration i, the only inner edges that may be visible are those that are incident to v_i and to no other vertex on the outer face.

(I4) *At the end* of each iteration (after removing the vertices that are not visible any more and before picking a new one), only vertices and edges incident to the outer face are visible.

Obviously, if the invariants hold, the set of visible edges in each frame forms a forest. In order to guarantee that the invariants hold, we use the following rules that determine which vertices we may not pick; see Figure 15. We call a vertex observing these rules *good*. Note that we always pick a good vertex on the outer face of G_i' – we will later argue that there is always one.

Rule 1: Do not pick a vertex v whose closed neighborhood $N[v] = \{v\} \cup \{u\colon uv \in E(G_i)\}$ contains all invisible vertices of the outer face of G_i'.

Rule 2: Do not pick an endpoint of a chord.

Rule 3: Do not pick a neighbor of an endpoint of a chord if the other endpoint of that chord is visible.

Rule 4: Do not pick an endpoint of a half-chord if the other endpoint is visible.

Rule 1 ensures that we do not close a cycle on the outer face, thus, invariant 1 holds. Rule 4 ensures that none of the visible inner vertices is adjacent to two visible vertices on the outer face (including the picked vertex), thus, invariant 2 holds. Rules 2 and 3 ensure that no chords are visible. Together with rule 4 and the fact that G is △-free, they ensure that the inner edges that are visible are incident to the picked vertex and no other vertex on the outer face. Thus, invariant 3 holds. Invariant 4 holds because we always pick a vertex on the outer face and remove it. As a result, the faces incident to the picked vertex become part

of the outer face and the previously inner neighbors (if any) of the picked vertex become incident to the outer face.

It remains to prove that, as long as G_i is not a forest (and the algorithm terminates), there exists a vertex that can be picked without violating any of our rules. Our proof is constructive; we show how to find a vertex to pick.

We first show that H_i is a (collection of) cactus graph(s), that is, every edge of H_i lies on at most one cycle. Suppose that H_i contains an edge e_1 that lies on at least two simple cycles. If the interiors of the two cycles are disjoint, then e_1 is not incident to the outer face of H_i (contradicting H_i being outerplane). Otherwise, one of the cycles has at least one edge $e_2 \neq e_1$ in the interior of the other cycle, again contradicting H_i being outerplane.

We show in two steps that G_i (actually even G'_i) always contains a good vertex, which we pick. First, we show how to find a good vertex in the base case, that is, if the outer face of G'_i is a simple cycle. Then, we consider the general case where the outer face of G'_i is a (collection of) cactus graph(s). Here, we repeatedly apply the argument of the base case to find a good vertex. So, assume that the outer face of G'_i is a simple cycle and, hence, H_i is connected.

In the trivial case that the weak dual H_i is a single vertex, G'_i is a cycle of at least four free vertices. Due to invariant 1, there is an invisible vertex $v \in G'_i$. Any non-neighbor of v in G'_i is a good vertex, which we can pick.

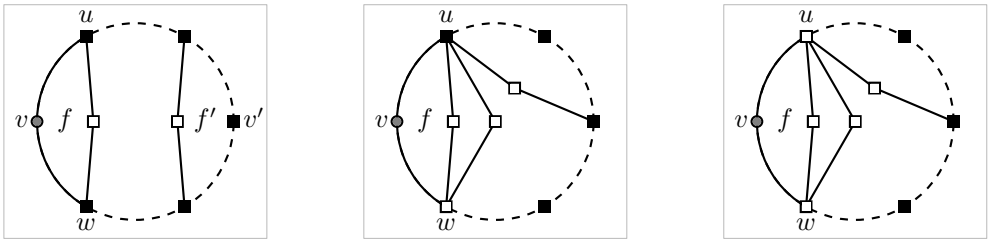
If H_i has a vertex of degree 1, which corresponds to a face f of G'_i, it means that f is incident to exactly one chord and to no half-chords. Since G is $\triangle$-free, there are at least two free vertices in f. Note that at most one endpoint of the chord is visible (due to invariant 3). If one endpoint is indeed visible, then its unique neighbor on the boundary of f that is not incident to the chord observes all rules and can be picked. If none of the endpoints of the chord is visible, then any free vertex of f can be picked.

Otherwise, all vertices of H_i have degree at least 2. Let F be the set of faces of G'_i that are incident to exactly one half-chord and to an arbitrary number of outer edges (but to no other inner edge). Note that in H_i, F corresponds to a set of vertices of degree 2. We now use the following two helpful claims.

▷ Claim 20. The set F has cardinality at least 2.

Proof. Consider the *block-cut tree* of H_i, that is, the tree that has a node for each cut vertex of H_i and a node for each 2-connected component (called *block*) of H_i. A block node and a cut-vertex node are connected by an edge in the block-cut tree if, in H_i, the block contains the cut vertex. Clearly, every leaf of the block-cut tree is a block node, and every block of H_i is either a cycle or an edge. Consider any leaf of the block-cut tree. It is a block node representing a cycle of H_i as otherwise it would represent an edge of H_i having an endpoint that is a vertex of degree 1 in H_i. Clearly, there is an inner vertex of G'_i being incident to all faces represented by that cycle and all edges of that cycle are then dual to half-chord edges of G'_i. As each such cycle contains at least two vertices but at most one of them is a cut vertex in H_i, there is, per leaf of the block-cut tree, at least one vertex of degree two in H_i whose incident edges are both dual to a half-chord of G'_i. The block-cut tree is either a single block node representing a cycle in H_i, or the block-cut tree has at least two leaf nodes. Thus, $|F| \geq 2$. ◁

▷ Claim 21. Let the edge e (or the edge pair $\{e_1, e_2\}$) be any chord (or half-chord) of G'_i, let F_1 be the set of inner faces on the one side, and let F_2 be the set of inner faces on the other side of e (or $\{e_1, e_2\}$, resp.). Then, $F_1 \cap F \neq \emptyset$ and $F_2 \cap F \neq \emptyset$.



(a) case 1 **(b)** case 2 **(c)** case 3

Figure 16 Cases when there is no chord in G_i'. We always find a good vertex.

Proof. If we have a chord e, then the dual edge of e is an edge of H_i, which corresponds to a block node of the block-cut tree, which cannot be a leaf (because all leaves represent cycles). Then, however, if we traverse the block-cut tree, in the one or the other direction of the edge, we will find a leaf in both parts and each leaf contains a vertex in F as shown in the proof of Claim 20.

If we have a half-chord $\{e_1, e_2\}$, then the dual edges divide a cycle C of H_i into two. If one of the resulting parts of C does not contain a cut vertex, then it contains only vertices from F by definition. Otherwise, the previous argument applies: the block-cut tree gets divided into two parts and each part needs to contain a leaf of the block-cut tree. ◁

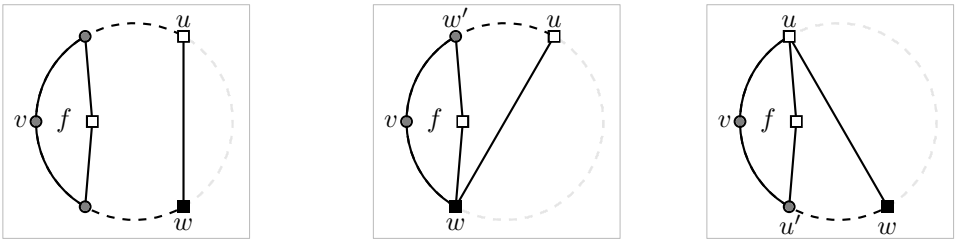
We continue to show that there is a good vertex on the outer face of G_i', which we can pick. Assume first that G_i' does not have chords. Thus, all vertices of G_i' trivially observe rules 2 and 3. Let $f \in F$, and let u and w be the endpoints of the unique half-chord incident to f. If there is a free vertex v in f such that $N[v]$ does not contain the last invisible vertices of the outer face of G_i', then we pick v. Rules 1 and 4 are observed by the definition of v. If, for every face $f \in F$ and for every free vertex v in f, $N[v]$ contains all invisible vertices of the outer face of G_i', consider the following three cases; see Figure 16. The cases are ordered by priority; if we fulfill the conditions of multiple cases, the first case applies.

(C1) Both u and w are visible. Then, consider a face $f' \in F$ different from f, which exists by Claim 20. Clearly, all of its vertices are visible, and we can pick any free vertex v' of f' without violating the rules.

(C2) Exactly one of $\{u, w\}$ is visible. Without loss of generality, assume that u is visible and w is invisible. We claim that u observes all rules. Since w remains invisible after picking u, u observes rule 1. If there was another half-chord incident to u, either it would again be incident to w, which does not violate rule 4, or it would be incident to another vertex of G_i', which is visible. By Claim 21, however, there is another face $f' \in F$ on the other side of that half-chord. As all of the vertices of f' on the outer face are visible, we would be in case 1 instead.

(C3) Both u and w are invisible. We claim that u observes the rules. Similar to case 2, u observes rule 1 (since w stays invisible) and rule 4 (if there was another half-chord incident to u whose other endpoint is visible, we would be in case 1 or in case 2).

Now assume that G_i' has one or more chords. Of course, each of these chords has at most one visible endpoint. The chords with exactly one visible endpoint divide G_i' into several subgraphs. Observe that at least one of these subgraphs contains no such chord in its interior and is bounded by only one of them (or no chord has a visible endpoint, then there is only one subgraph, namely G_i'). We call this subgraph $\hat{G}_i'$ and we let u and w denote the visible



(a) f is not incident to u or w. **(b)** f is incident to w. **(c)** f is incident to u.

Figure 17 Cases when there are chords in G'_i. We always find a good vertex.

and invisible endpoints of the bounding chord e, respectively (or if there is only one subgraph, then u and w are just neighbors). By a case distinction on the faces incident to u and w, we can show that there is always a good vertex on the outer face of $\hat{G}'_i$, and hence on the outer face of G'_i.

Let $\hat{F}$ be the subset of F that is contained in $\hat{G}'_i$, which is non-empty by Claim 21. If there is a face $f \in \hat{F}$ neither incident to u nor w, then we can pick a free vertex v of f; see Figure 17a. Since at least u stays invisible, v observes rule 1. Trivially, the other rules are also observed.

Otherwise all faces in $\hat{F}$ are incident either to u or to w. Note that none of the faces in $\hat{F}$ can be incident to both u and w as this would imply a triangle. Assume that there is a face $f \in \hat{F}$ incident to a half-chord with endpoints w and w'; see Figure 17b. We claim that we can pick a free vertex v of f. Since at least u stays invisible, v observes rule 1. Rules 2 and 4 are trivially observed. Rule 3 is also observed, since w' is not incident to a chord whose other endpoint is visible (as they are all inside $\hat{G}'_i$) and w is not incident to a chord whose other endpoint is visible, otherwise invariant 3 would be violated.

Finally, all faces in $\hat{F}$ are incident to u. For a face $f \in \hat{F}$, let u' be the other endpoint of the half-chord; see Figure 17c. Note that it might happen now that all free vertices of f break rule 1 or rule 3. We claim, however, that we can always pick u'. Because of u, u' observes rule 1. Furthermore, u' observes rule 2 and rule 4 as u' cannot be incident to a chord or another half-chord distinct from the half-chord $\{u, u'\}$. If u' was incident to a chord or a half-chord, this chord or half-chord would divide $\hat{G}'_i$ into two parts and then there would be a face in $\hat{F}$ that is not incident to u due to Claim 21. By the same argument, all chords and half-chords of $\hat{G}'_i$ are incident to u. In order to show that u' observes rule 3, note that the only vertex of $\hat{G}'_i$ that is incident to a chord whose other vertex is visible is u. Since u' is adjacent neither to u (otherwise there is a triangle) nor to any vertex that is not in $\hat{G}'_i$, u' observes rule 3 as well and, hence, is a good vertex.

We have shown that there is always a good vertex on the outer face of G'_i if the outer face of G'_i is a simple cycle. Now assume that the outer face of G'_i is not just a simple cycle, but consists of one or multiple cactus graphs. If we have multiple cactus graphs, we can consider them individually. So, it suffices to consider the case where the outer face of G'_i is one (connected) cactus graph. Still, H_i may be disconnected. Let $C_1, C_2, \dots$ be the connected components of H_i, and let $\tilde{G}_1, \tilde{G}_2, \dots$ be the corresponding subgraphs of G'_i. Two subgraphs $\tilde{G}_j$ and $\tilde{G}_k$ may be connected by at most one common vertex or via a single edge. Otherwise, we consider them as non-connected (if they are connected by a path of length ≥ 2 in G_i, they are independent because the neighborhood of $\tilde{G}_j$ does not overlap $\tilde{G}_k$ and vice versa; these parts remain as a forest in the end). Let T be a graph with a vertex for each $\tilde{G}_1, \tilde{G}_2, \dots$ where two vertices are adjacent if and only if the corresponding subgraphs

are connected. Since the outer face of G'_i is a cactus graph, T is a forest. Consider the subgraph $\tilde{G}_1$ and use the algorithm above to find a good vertex v. If v is a cut vertex, then check if it is also a good vertex in all subgraphs from $\{\tilde{G}_1, \tilde{G}_2, \dots\}$ where it is contained as well. Further, check for each neighbor w of v that is contained in a subgraph $\tilde{G}_j$ distinct from $\tilde{G}_1$ whether making w visible violates one of the invariants (note that this is a weaker criterion than checking if w is a good vertex and it implies that w and its neighbors in $\tilde{G}_j$ are not good vertices). If there is a subgraph $\tilde{G}_j$ where picking v breaks at least one rules (or making a neighbor of v visible breaks an invariant), then find a good vertex in $\tilde{G}_j$ (recall that there exists at least one good vertex) and proceed in the same way. Since T does not contain cycles, this procedure always terminates with a (globally) good vertex.

Our constructive proof can be turned into a polynomial-time algorithm. Let n be the number of vertices of G.

▷ Claim 22. The algorithm runs in $O(n^2)$ time.

Proof. If Γ is not given, we can compute it in linear time [13, 22]. We mark the vertices that are currently on the outer face. This can be done initially in linear time. Updating the vertices on the outer face can be done in linear total time as every vertex disappears at most once. To keep track when a vertex disappears, we maintain, for each vertex, a counter of the neighbors that have not been visible yet. Whenever a vertex appears, we update this counter over the incident edges. As we have a linear number of edges and at most two updates per edge, this can also be done in linear total time.

We next describe how to represent G'_i. For each of the two sides of every edge, we keep a flag that is true if and only if the side is incident to the outer face. The flags can be initialized in linear time. Whenever a vertex disappears, we traverse the faces that have become part of the outer face and update the corresponding flags; as every edge side becomes incident to the outer face at most once, this step can be done in linear total time. Using these flags allows us to keep track of the vertices on the outer face of G'_i and the inner vertices (half-chord vertices) of G'_i. We maintain a list L of the vertices on the outer face of G'_i. At the same time, for each vertex, we maintain a mark that is true if the vertex lies on the outer face of G'_i. A vertex v lies on the outer face of G'_i if v is on the current outer face (according to its mark) and if at least one of the edges incident to v has a flag saying that one of its sides is not part of the outer face. The list L and the corresponding flags can be initialized for G'_1 in linear time. Then, we can mark in linear time the inner vertices of G'_1 by checking, for every vertex v that is not marked to be on the current outer face, whether v has two or more neighbors on the outer face of G'_1. We update L and all flags marking vertices of G'_i whenever a vertex disappears. When updating, we also maintain, for each inner vertex of G'_i, how many of its neighbors on the outer face of G'_i are visible. All of this can be done in linear time per iteration, and in quadratic total time. Note that we do not explicitly mark the edges of G'_i. For each edge e, we can check in constant time whether both endpoints of e belong to G'_i and at least one of them lies on the outer face – then e belongs to G'_i.

Finally, we discuss the running time for finding a good vertex, which we pick in the next iteration. By definition, a vertex is good if it observes rules 1–4. We can simply iterate over each vertex v in L, and check in $\deg(v)$ time whether it observes all rules. In particular, note that, for the inner half-chord vertices neighboring v (relevant for rule 4), we have a counter saying how many of its neighbors on the outer face of G'_i are visible (and we know whether v is visible itself). We have shown that there always exists a good vertex in L. In the worst case, these checks (and hence each iteration) take linear total time. Since the number of iterations is linear, the algorithm runs in quadratic total time. ◁

This finishes the proof of the theorem. ◀

8 Open problems

1. It is interesting to study further parameterizations of Forest Storyplan and Outerplanar Storyplan. Is there an FPT algorithm with respect to the treewidth (pathwidth) of the given graph?
2. While we extended the existing planar storyplan problem into the direction of less powerful but easier-to-understand storyplans, one could also go into the opposite direction and investigate more powerful storyplans in order to be able to construct such storyplans for larger classes of graphs. For example, 1-planar storyplans would be a natural direction for future research. It turns out, however, that deciding whether a given graph admits a 1-planar storyplan is also NP-hard [23]. Hence, it would be interesting to see whether the problem admits a parameterized approach for this class of graphs.
3. What are the smallest graphs in terms of the number of vertices or edges that do not admit forest storyplans, outerplanar storyplans, planar storyplans, or 1-planar storyplans?

References

1 Moataz Abdelaal, Antoine Lhuillier, Marcel Hlawatsch, and Daniel Weiskopf. Time-aligned edge plots for dynamic graph visualization. In Ebad Banissi et al., editor, *Proc. 24th Int. Conf. Inform. Vis. (IV)*, pages 248–257, 2020. doi:10.1109/IV51561.2020.00048.

2 Ken Been, Martin Nöllenburg, Sheung-Hung Poon, and Alexander Wolff. Optimizing active ranges for consistent dynamic map labeling. In Monique Teillaud, editor, *Proc. 24th ACM Symp. Comput. Geom. (SoCG)*, pages 10–19, 2008. doi:10.1145/1377676.1377681.

3 Ken Been, Martin Nöllenburg, Sheung-Hung Poon, and Alexander Wolff. Optimizing active ranges for consistent dynamic map labeling. *Comput. Geom.*, 43(3):312–328, 2010. doi:10.1016/j.comgeo.2009.03.006.

4 Carla Binucci, Ulrik Brandes, Giuseppe Di Battista, Walter Didimo, Marco Gaertler, Pietro Palladino, Maurizio Patrignani, Antonios Symvonis, and Katharina Anna Zweig. Drawing trees in a streaming model. *Inf. Process. Lett.*, 112(11):418–422, 2012. doi:10.1016/j.ipl.2012.02.011.

5 Carla Binucci, Emilio Di Giacomo, William J. Lenhart, Giuseppe Liotta, Fabrizio Montecchiani, Martin Nöllenburg, and Antonios Symvonis. On the complexity of the storyplan problem. In Patrizio Angelini and Reinhard von Hanxleden, editors, *Proc. 30th Int. Symp. Graph Drawing & Network Vis. (GD)*, volume 13764 of *LNCS*, pages 304–318. Springer, 2023. URL: https://arxiv.org/abs/2209.00453, doi:10.1007/978-3-031-22203-0_22.

6 Carla Binucci, Emilio Di Giacomo, William J. Lenhart, Giuseppe Liotta, Fabrizio Montecchiani, Martin Nöllenburg, and Antonios Symvonis. On the complexity of the storyplan problem. *J. Comput. Syst. Sci.*, 139:103466:1–17, 2024. doi:10.1016/J.JCSS.2023.103466.

7 Manuel Borrazzo, Giordano Da Lozzo, Giuseppe Di Battista, Fabrizio Frati, and Maurizio Patrignani. Graph stories in small area. *J. Graph Algorithms Appl.*, 24(3):269–292, 2020. doi:10.7155/jgaa.00530.

8 Manuel Borrazzo, Giordano Da Lozzo, Fabrizio Frati, and Maurizio Patrignani. Graph stories in small area. In Daniel Archambault and Csaba D. Tóth, editors, *Proc. 27th Int. Symp. Graph Drawing and Network Visualization (GD)*, volume 11904 of *LNCS*, pages 545–558. Springer, 2019. URL: https://arxiv.org/abs/1908.09318, doi:10.1007/978-3-030-35802-0_41.

9 Michael Burch. The dynamic graph wall: Visualizing evolving graphs with multiple visual metaphors. *J. Vis.*, 20(3):461–469, 2017. doi:10.1007/s12650-016-0360-z.

10 Jianer Chen, Iyad A. Kanj, and Ge Xia. Improved upper bounds for vertex cover. *Theor. Comput. Sci.*, 411(40-42):3736–3756, 2010. doi:10.1016/J.TCS.2010.06.026.

11 Giordano Da Lozzo and Ignaz Rutter. Planarity of streamed graphs. In Vangelis Th. Paschos and Peter Widmayer, editors, *Proc. 9th Int. Conf. Algorithms & Complexity (CIAC)*, volume 9079 of *LNCS*, pages 153–166. Springer, 2015. URL: https://arxiv.org/abs/1501.07106, doi:10.1007/978-3-319-18173-8_11.

12 Giordano Da Lozzo and Ignaz Rutter. Planarity of streamed graphs. *Theor. Comput. Sci.*, 799:1–21, 2019. doi:10.1016/j.tcs.2019.09.029.

13 Hubert de Fraysseix, János Pach, and Richard Pollack. How to draw a planar graph on a grid. *Combin.*, 10(1):41–51, 1990. doi:10.1007/BF02122694.

14 Giuseppe Di Battista, Walter Didimo, Luca Grilli, Fabrizio Grosso, Giacomo Ortali, Maurizio Patrignani, and Alessandra Tappini. Small point-sets supporting graph stories. In Patrizio Angelini and Reinhard von Hanxleden, editors, *Proc. 30th Int. Symp. Graph Drawing & Network Vis. (GD)*, volume 13764 of *LNCS*, pages 289–303. Springer, 2023. URL: https://arxiv.org/abs/2208.14126, doi:10.1007/978-3-031-22203-0_21.

15 Giuseppe Di Battista, Walter Didimo, Luca Grilli, Fabrizio Grosso, Giacomo Ortali, Maurizio Patrignani, and Alessandra Tappini. Small point-sets supporting graph stories. *J. Graph Algorithms Appl.*, 27(8):651–677, 2023. doi:10.7155/JGAA.00639.

16 Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, 1999.

17 Giuseppe Di Battista and Fabrizio Frati. Small area drawings of outerplanar graphs. In Patrick Healy and Nikola S. Nikolov, editors, *Proc. 13th Int. Symp. Graph Drawing (GD)*, volume 3843 of *LNCS*, pages 89–100. Springer, 2005. doi:10.1007/11618058_9.

18 Giuseppe Di Battista and Fabrizio Frati. Small area drawings of outerplanar graphs. *Algorithmica*, 54(1):25–53, 2009. doi:10.1007/11618058_9.

19 Alexander Dobler, Maximilian Holzmüller, and Martin Nöllenburg. Geometry matters in planar storyplans. In Vida Dujmović and Fabrizio Montecchiani, editors, *Proc. 33rd Int. Symp. Graph Drawing & Network Vis. (GD)*, volume 357 of *LIPIcs*, pages 27:1–27:9. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.GD.2025.27.

20 Jiří Fiala, Oksana Firman, Giuseppe Liotta, Alexander Wolff, and Johannes Zink. Outerplanar and forest storyplans. In Henning Fernau, Serge Gaspers, and Ralf Klasing, editors, *Proc. 49th Int. Conf. Current Trends Theory & Practice Comput. Sci. (SOFSEM)*, volume 14519 of *LNCS*, pages 211–225. Springer, 2024. doi:10.1007/978-3-031-52113-3_15.

21 Thomas J. Schaefer. The complexity of satisfiability problems. In *Proc. 10th Annu. ACM Symp. Theory Comput. (STOC)*, pages 216–226, 1978. doi:10.1145/800133.804350.

22 Walter Schnyder. Embedding planar graphs on the grid. In *Proc. 1st ACM-SIAM Symp. Discrete Algorithms (SODA)*, pages 138–148, 1990. URL: https://dl.acm.org/doi/10.5555/320176.320191.

23 Jan Sulejmani. 1-planar storyplans. Bachelor's thesis, 2024. URL: https://www1.pub.informatik.uni-wuerzburg.de/pub/theses/2024-sulejmani-bachelor.pdf.