











On the Complexity of the Storyplan Problem

Carla Binucci¹ , Emilio Di Giacomo¹  , William J. Lenhart² ,
Giuseppe Liotta¹ , Fabrizio Montecchiani¹ , Martin Nöllenburg³ ,
and Antonios Symvonis⁴ 

¹ Department of Engineering, University of Perugia, Perugia, Italy
{carla.binucci,emilio.digiacomo,giuseppe.liotta,
fabrizio.montecchiani}@unipg.it

² Department of Computer Science, Williams College, Williamstown, USA
wlenhart@williams.edu

³ Algorithms and Complexity Group, TU Wien, Vienna, Austria
noellenburg@ac.tuwien.ac.at

⁴ School of Applied Mathematical and Physical Sciences, NTUA, Athens, Greece
symvonis@math.ntua.gr

Abstract. Motivated by dynamic graph visualization, we study the problem of representing a graph G in the form of a *storyplan*, that is, a sequence of frames with the following properties. Each frame is a planar drawing of the subgraph of G induced by a suitably defined subset of its vertices. Between two consecutive frames, a new vertex appears while some other vertices may disappear, namely those whose incident edges have already been drawn in at least one frame. In a storyplan, each vertex appears and disappears exactly once. For a vertex (edge) visible in a sequence of consecutive frames, the point (curve) representing it does not change throughout the sequence.

Note that the order in which the vertices of G appear in the sequence of frames is a total order. In the STORYPLAN problem, we are given a graph and we want to decide whether there exists a total order of its vertices for which a storyplan exists. We prove that the problem is NP-complete, and complement this hardness with two parameterized algorithms, one in the vertex cover number and one in the feedback edge set number of G . Also, we prove that partial 3-trees always admit a storyplan, which can be computed in linear time. Finally, we show that the problem remains NP-complete in the case in which the total order of the vertices is given as part of the input and we have to choose how to draw the frames.

Keywords: Dynamic graph drawing · NP-hardness · Parameterized analysis · Pathwidth

Research partially supported by MIUR grant 20174LF3T8 “AHeAD: efficient Algorithms for HARnessing networked Data”, Progetto RICBA21LG “Algoritmi, modelli e sistemi per la rappresentazione visuale di reti”, and the Vienna Science and Technology Fund (WWTF) grant ICT19-035.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
P. Angelini and R. von Hanxleden (Eds.): GD 2022, LNCS 13764, pp. 304–318, 2023.
https://doi.org/10.1007/978-3-031-22203-0_22

1 Introduction

Let $G = (V, E)$ be a graph with n vertices. We write $[n]$ as shorthand for the set $\{1, 2, \dots, n\}$. A *storyplan* $\mathcal{S} = \langle \tau, \{D_i\}_{i \in [n]} \rangle$ of G is a pair defined as follows. The first element is a bijection $\tau : V \rightarrow [n]$ that represents a total order of the vertices of G . For a vertex $v \in V$, let $i_v = \tau(v)$ and let $j_v = \max_{u \in N[v]} \tau(u)$, where $N[v]$ is the set containing v and its neighbors. The *lifespan* of v is the interval $[i_v, j_v]$. We say that v *appears* at step i_v , is *visible* at step i for each $i \in [i_v, j_v]$, and *disappears* at step $j_v + 1$. Note that a vertex does not disappear until all its neighbors have appeared. The second element of \mathcal{S} is a sequence of drawings $\{D_i\}_{i \in [n]}$, such that: (i) each drawing D_i contains all vertices visible at step i , (ii) each drawing D_i is planar, (iii) the point representing a vertex v is the same over all drawings that contain v (i.e., it does not change during the lifespan of v), and (iv) the curve representing an edge e is the same over all drawings that contain e . We introduce the STORYPLAN problem.

STORYPLAN

Input: Graph $G = (V, E)$

Question: Does G admit a storyplan?

In what follows, each drawing D_i of a storyplan \mathcal{S} is called a *frame* of \mathcal{S} . Also, we denote by $|D_i|$ the number of vertices of D_i , while the *width* of \mathcal{S} is $w(\mathcal{S}) = \max_{i \in [n]} |D_i| - 1$ (we subtract one to align the definition with other width parameters). If G admits a storyplan, then the *framewidth* of G , denoted by $\text{fw}(G)$, is the minimum width over all its storyplans; otherwise the framewidth of G is conventionally set to $+\infty$. We will observe that the framewidth of G upper bounds its pathwidth [12], since each frame can be interpreted as a bag of a path decomposition with the addition of conditions (ii)–(iv).

Motivation and Related Work. Testing for the existence of a storyplan of a graph generalizes planarity and it is of theoretical interest as it combines classical width parameters of graphs with topological properties. From a more practical perspective, computing a storyplan (if any) of a graph G is a natural way to gradually visualize G in a story-like or small-multiples fashion, such that each single drawing is planar and the reader's mental map is preserved throughout the sequence of drawings (see, e.g., [8] for a similar approach). More in general, the problem of visualizing graphs that change over time has motivated a notable amount of literature in graph drawing and network visualization (see, e.g., [2, 3, 6, 7, 15, 16]). While numerous dynamic graph visualization models have been proposed, two works are of particular interest for our research. The first one is the work by Borrazzo et al. [6], in which the following problem is introduced. A *graph story* is formed by a graph G , a total order of its vertices τ , and a positive integer W . The problem is to find a sequence of drawings $\{D_i\}_{i \in [n]}$ in which each D_i contains all vertices v such that $i - W < \tau(v) \leq i$, and the position of a vertex is the same over all drawings it belongs to. Borrazzo et al. prove that any story of a path or a tree can be drawn on a $2W \times 2W$ and on an $(8W + 1) \times (8W + 1)$ grid, respectively, so that all the drawings of the story are

straight-line and planar. Note that having a fixed window of size W implies that at most $O(W \cdot n)$ edges of G can be represented, in particular, any edge whose endpoints are at distance larger than W in τ does not appear in any drawing. Having both a fixed order and a fixed lifespan are the key differences with our setting. In particular, unconstrained lifespans allow us to find stories in which all edges are drawn in at least one step, while planarity still guarantees that even large frames are readable. Besides such differences in the models, our focus is on the complexity of the decision problem, rather than on area bounds for specific graph families. The second work is by Da Lozzo and Rutter [7], who introduce *stream planarity*. Given a graph G , a total order τ of the *edges* of G , and a positive integer W , stream planarity asks for a sequence of drawings $\{D_i\}_{i \in [n]}$ in which each D_i contains all edges e such that $i - W < \tau(e) \leq i$, and the subdrawing of the vertices and edges shared by D_i and D_{i-1} is the same in both drawings. Da Lozzo and Rutter prove that there exists a constant value for W for which the stream planarity problem is NP-complete. They also study a variant where a backbone graph is given whose edges must stay in the drawing at each time step; for this variant they prove that the problem is NP-complete for all $W \geq 2$ and can be solved in polynomial time when $W = 1$ or when the backbone graph is biconnected. The difference of stream planarity with our problem, besides the fact that edges are streamed rather than vertices, is again having a fixed order and a fixed lifespan.

Contribution. The main results in this paper can be summarized as follows.

- We show that STORYPLAN is NP-complete (Sect. 3.1). As we reduce from ONE-IN-THREE 3SAT and we blow up the instance by a linear factor, it follows that there is no algorithm that solves STORYPLAN in $2^{o(n)}$ time unless ETH fails. On the other hand, such a lower bound can be complemented with a simple algorithm running in $2^{O(n \log n)}$ time.
- Motivated by the above hardness, we study the parameterized complexity of STORYPLAN and describe two fixed-parameter tractable algorithms. We first show that STORYPLAN belongs to FPT when parameterized by the vertex cover number via the existence of a kernel, whose size is however super-polynomial (Sect. 3.2). We then prove that STORYPLAN parameterized by the feedback edge set number (i.e., the minimum number of edges whose removal makes the graph acyclic) admits a kernel of linear size (Sect. 3.3).
- In parameterized analysis, a central parameter to consider is treewidth. In this direction, finding a parameterized algorithm for STORYPLAN appears to be an elusive task. However, we show that for partial 3-trees, a storyplan always exists and can be computed in linear time (Sect. 3.4).
- Finally, we initiate the study of the complexity of a variant of STORYPLAN in which the total order of the vertices is fixed in advance (but the vertex lifespan remains unconstrained). We prove NP-completeness for this problem via a reduction from SUNFLOWER SEFE [14] (Sect. 4).

Some proofs are omitted and can be found in [4]; the corresponding statements are marked (\star).

2 Preliminaries and Basic Results

A *drawing* Γ of a graph $G = (V, E)$ is a mapping of the vertices of V to points in the plane \mathbb{R}^2 , and of the edges of E to Jordan arcs connecting their corresponding endpoints but not passing through any other vertex. Drawing Γ is *planar* if no edge is crossed. A graph is *planar* if it admits a planar drawing. A planar drawing of a planar graph G subdivides the plane into topologically connected regions, called *faces*. The infinite region is the *outer face*. A *planar embedding* \mathcal{E} of G is an equivalence class of planar drawings that define the same set of faces and the same outer face. For any $V' \subseteq V$, we denote by $G[V']$ the subgraph of G induced by the vertices of V' and by $\Gamma[V']$ the subdrawing of Γ representing $G[V']$.

Connection with Pathwidth. The next properties show some simple connections between storyplans and path decompositions [12].

Theorem 1 (\star). *Let $G = (V, E)$ be a graph, then $\text{pw}(G) \leq \text{fw}(G)$. Also, if G is planar then it always admits a storyplan, and in particular $\text{pw}(G) = \text{fw}(G)$.*

Since computing the pathwidth is NP-hard already for planar graphs of bounded degree [11], the next corollary immediately follows from Theorem 1.

Corollary 1. *Computing the framewidth of a graph is NP-hard for planar graphs of bounded degree.*

Analogously, computing the pathwidth of a graph is FPT in the pathwidth [5], hence computing the framewidth of a planar graph is also FPT in the framewidth.

Complete Bipartite Graphs. It is not difficult to verify that if a graph admits a storyplan, then it does not contain K_5 as a subgraph. However, complete bipartite graphs always admit a storyplan and such storyplans have important properties. The next statement plays a central role in most of our proofs.

Lemma 1 (\star). *Let $K_{a,b} = (A \cup B, E)$ be a complete bipartite graph with $a = |A|$, $b = |B|$, and $3 \leq b \leq a$. Let $\mathcal{S} = \langle \tau, \{D_i\}_{i \in [a+b]} \rangle$ be a storyplan of $K_{a,b}$. Exactly one of A and B is such that all its vertices are visible at some $i \in [a+b]$.*

In view of Lemma 1, we have the following definition.

Definition 1. *For a complete bipartite graph $K_{a,b}$ with $3 \leq b \leq a$ and a storyplan \mathcal{S} of $K_{a,b}$, we call fixed the partite set of $K_{a,b}$ whose vertices are all visible at some step of \mathcal{S} , and flexible the other partite set.*

3 Complexity of STORYPLAN

In this section we prove that: STORYPLAN is NP-complete and cannot be solved in $2^{o(n)}$ time unless ETH fails, but there is an algorithm running in $2^{O(n \log n)}$ time (Sect. 3.1); STORYPLAN is in FPT parameterized by vertex cover number or feedback edge set number (Sects. 3.2 and 3.3); graphs of treewidth at most 3 always admit a storyplan, which can be computed in linear time (Sect. 3.4).

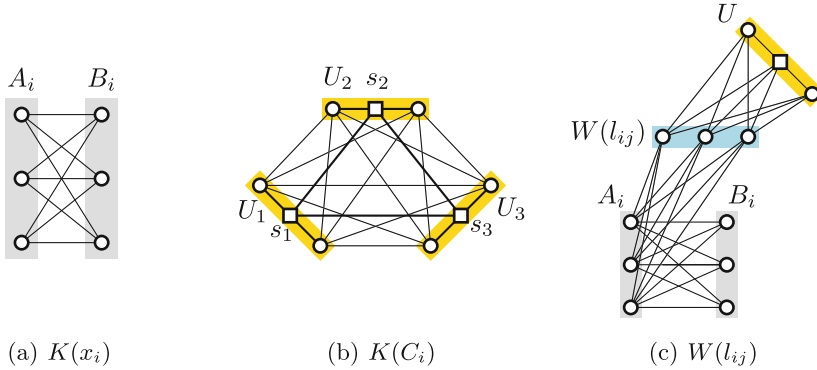


Fig. 1. Illustration for the reduction of Theorem 2.

3.1 Hardness

We reduce from ONE-IN-THREE 3SAT, a variant of 3SAT which asks whether there is a satisfying assignment in which *exactly* one literal in each clause is true. Let φ be a 3SAT formula over N variables $\{x_i\}_{i \in [N]}$ and M clauses $\{C_i\}_{i \in [M]}$. We construct an instance of STORYPLAN, i.e., a graph $G = (V, E)$, as follows; refer to Fig. 1 for an illustration.

Variable Gadget. Each variable x_i is represented in G by a copy $K(x_i)$ of $K_{3,3}$ (see Fig. 1(a)). Let A_i and B_i be the two partite sets of $K(x_i)$, which we call the *v-sides* of $K(x_i)$. A true (false) assignment of x_i will correspond to set A_i being flexible (fixed) in a putative storyplan of G (see Definition 1).

Clause Gadget. Consider a copy of $K_{2,2,2} = (U_1 \cup U_2 \cup U_3, F)$. An *extended* $K_{2,2,2}$ is the graph obtained from any such a copy by adding three vertices s_1, s_2, s_3 , such that these three vertices are pairwise adjacent, and each s_j is adjacent to both vertices in U_j , for $j \in \{1, 2, 3\}$. In the following, s_1, s_2, s_3 are the *special vertices* of the extended $K_{2,2,2}$, while the other vertices are the *simple vertices*. A clause C_i is represented in G by an extended $K_{2,2,2}$, denoted by $K(C_i)$ (see Fig. 1(b)). In particular, we call each of the three sets of vertices $U_j \cup \{s_j\}$ a *c-side* of $K(C_i)$. The idea is that $K(C_i)$ admits a storyplan if and only if exactly one c-side is flexible (each c-side will be part of a $K_{3,3}$, see the wire gadget below).

Wire Gadget. Refer to Fig. 1(c). Let x_i be a variable having a literal l_{ij} in a clause C_j . Any such variable-clause incidence is represented in G by a set of three vertices, which we call the *w-side* $W(l_{ij})$. All vertices of $W(l_{ij})$ are connected to all vertices of one of the three c-sides of $K(C_j)$, which we call U , such that the graph induced by $W(l_{ij}) \cup U$ in G contains a copy of $K_{3,3}$. Also, each vertex of $W(l_{ij})$ is connected to all vertices of the v-side A_i (B_i) if the literal is positive (negative), such that the graph induced by $W(l_{ij}) \cup A$ ($W(l_{ij}) \cup B$) in G is a copy of $K_{3,3}$. Also, note that each c-side of $K(C_j)$ is adjacent to exactly one w-side.

Lemma 2 (★). *If graph G admits a storyplan then φ admits a satisfying assignment with exactly one true literal in each clause.*

Proof (Sketch). Let \mathcal{S} be a storyplan of G . For each variable gadget $K(x_i)$ we assign the value *true* to x_i if the v-side A_i is flexible in \mathcal{S} . Consider any literal l_{ij} and the wire gadget W_{ij} . If l_{ij} is positive (negative), then A_i (B_i) and W_{ij} form a $K_{3,3}$, hence by Lemma 1 the w-side W_{ij} is fixed (flexible). Analogously, if we consider the clause gadget $K(C_j)$, the c-side connected with W_{ij} is flexible (fixed). Symmetrically, we assign the value *false* to x_i if the v-side B_i is instead flexible in \mathcal{S} , and for any positive (negative) literal l_{ij} , the w-side W_{ij} is flexible (fixed), while the corresponding c-side of $K(C_j)$ is fixed (flexible). In other words, the value of x_i propagates consistently throughout all its literals. It remains to prove that, for any clause C_j of φ , precisely one literal is true. Namely, we claim that exactly one c-side of $K(C_j)$ is flexible, while the other two are fixed. At high level, we rely on the fact that an extended $K_{2,2,2}$ wants at least two c-sides to be fixed, while the special vertices force at least one c-side to be flexible.

Lemma 3 (★). *If the formula φ admits a satisfying assignment with exactly one true literal in each clause, then graph G admits a storyplan.*

Proof (Sketch). Given a satisfying assignment of φ with one true literal per clause, we can compute a storyplan $\mathcal{S} = \langle \tau, \{D_i\}_{i \in [n]} \rangle$ of G . In what follows, when the order of a group of vertices is not specified, any relative order is valid.

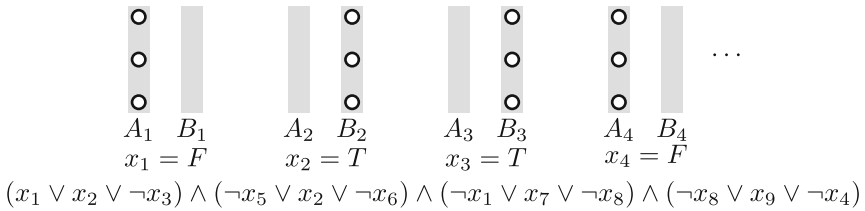


Fig. 2. Proof of Lemma 3: drawing the vertices of the fixed v-sides.

Consider a single variable gadget $K(x_i)$. If x_i is true in the satisfying assignment, then we let appear the three vertices of the v-side B_i of $K(x_i)$, that is, B_i is the fixed side of $K(x_i)$. If x_i is false, we do the opposite, namely we let appear the three vertices of the v-side A_i of $K(x_i)$. This procedure is repeated for all variables in any order. For ease of presentation, we can imagine that all the drawn v-sides are horizontally aligned, as shown in Fig. 2. Thus, for the variable gadgets, it remains to draw their flexible v-sides.

Consider now a wire gadget $W(l_{ij})$. If x_i is true and l_{ij} is positive, then $W(l_{ij})$ must be fixed because it forms a $K_{3,3}$ with the v-side A_i of $K(x_i)$, which is flexible. Therefore we let appear the three vertices of $W(l_{ij})$. Similarly, if x_i is false and l_{ij} is negative, then $W(l_{ij})$ must be fixed, and we let appear the three vertices of $W(l_{ij})$. Again, this procedure is repeated for all wires in any order.

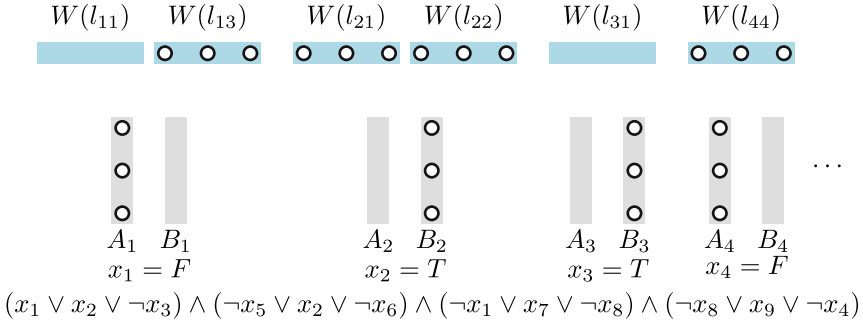


Fig. 3. Proof of Lemma 3: drawing the vertices of the fixed w-sides.

For ease of presentation, we can imagine that all the drawn w-sides are arranged along a horizontal line slightly above the variable gadgets, as shown in Fig. 3. Thus, also for the wire gadgets, it remains to draw the flexible w-sides.

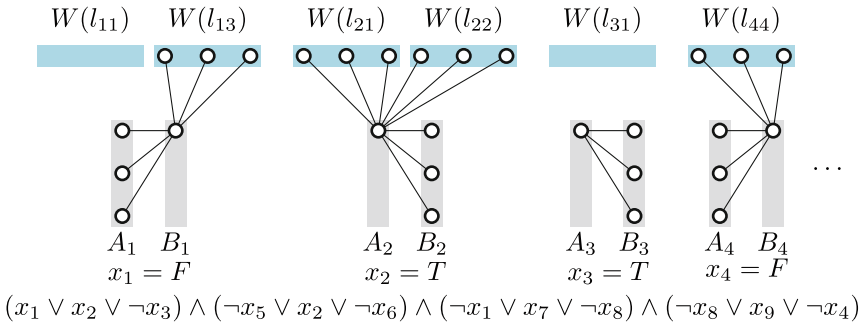


Fig. 4. Proof of Lemma 3: drawing the vertices of the flexible v-sides.

We sketch the remaining part of the proof (see [4] for a full proof). Flexible v-sides can be drawn as in Fig. 4. Figure 5 shows how to draw a clause gadget, ignoring the connections with the linked wire gadgets. Finally, in order to draw the flexible w-sides and their edges, and the edges between the fixed w-sides and the corresponding c-sides, we enclose all the wire and variable gadgets in a face of the current clause gadget where all vertices of the linked c-side are visible.

Theorem 2 (*). *The STORYPLAN problem is NP-hard and it has no $2^{o(n)}$ time algorithm unless ETH fails.*

The above lower bound for the running time of an algorithm solving STORYPLAN can be easily complemented with a nearly tight upper bound. The proof of the next theorem also shows that STORYPLAN belongs to NP. Namely, it gives a nondeterministic scheme to generate a set of candidate solutions, and then it

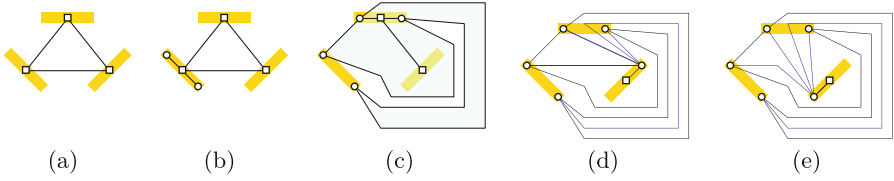


Fig. 5. Proof of Lemma 3: drawing a clause gadget.

shows how to check, in polynomial time, if a candidate solution is valid. However, it intertwines such process in order to obtain a lower time complexity.

Theorem 3 (\star). *The STORYPLAN problem is in NP. Also, given an n -vertex graph G , there is an algorithm that solves STORYPLAN on G in $2^{O(n \log n)}$ time.*

Proof (Sketch). We first guess a total order of the vertices of G . This fixes for each $i \in [n]$ the visible vertices. Next, for each $i \in [n]$, we generate the possible planar embeddings (rather than planar drawings) of the graph induced by the vertices visible at step i , and discard any embedding \mathcal{E} for which there is no planar embedding \mathcal{E}' generated at step $i - 1$ (if $i > 1$) such that the restrictions of \mathcal{E} and \mathcal{E}' to the common subgraph coincide. If the algorithm returns at least one planar embedding at step n , there is a sequence of planar embeddings in which common subgraphs share the same embedding, hence G admits a storyplan.

3.2 Parameterization by Vertex Cover Number

A *vertex cover* of a graph $G = (V, E)$ is a set $C \subseteq V$ such that every edge of E 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 prove the following by means of kernelization.

Theorem 4 (\star). *Let $G = (V, E)$ be a graph with n vertices and vertex cover number $\kappa = \kappa(G)$. Deciding whether G admits a storyplan, and computing one if any, can be done in $O(2^{2^{O(\kappa)}} + n^2)$ time.*

Algorithm Description. Without loss of generality, we assume that the input graph G does not contain isolated vertices, as they do not affect the existence of a storyplan. Let C be a vertex cover of size $\kappa = \kappa(G)$ of graph 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 $\sum_{i=1}^{\kappa} \binom{\kappa}{i} = 2^{\kappa} - 1 < 2^{\kappa}$ distinct types. Denote by V_U the set of vertices of type U . We define three reduction rules.

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

R.2: 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 .

R.3: 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 .

Lemma 4 (★). Let G' be the graph obtained from G by applying one of the reduction rules **R.1–R.3**. Then G admits a storyplan if and only if G' does.

Proof (Sketch). For the nontrivial direction, suppose that G' admits a storyplan $\mathcal{S}' = \langle \tau', \{D'_i\}_{i \in [n']}\rangle$, where $n' = n - 1$. We can distinguish three cases based on the reduction rule applied to G . Here we only prove the simplest of the three cases, namely the case in which **R.1** is applied. See Fig. 6 for an illustration. Let x be the vertex removed from G to obtain G' and let v be its neighbor, whose lifespan according to τ' is $[i_v, j_v]$. We compute τ from τ' by inserting x right after v , thus the lifespan of x in τ is $[i_v + 1, i_v + 1]$. Similarly, we compute $\{D_i\}_{i \in [n]}$ from $\{D'_i\}_{i \in [n']}$ as follows. For each $i \leq i_v$, we set $D_i = D'_i$. For $i = i_v + 1$, we draw x in D'_{i_v} sufficiently close to v such that edge xv can be drawn as a straight-line segment that does not intersect any other edge. We then set D_i to be equal to the resulting drawing. For each $i > i_v + 1$, we set $D_i = D'_{i-1}$.

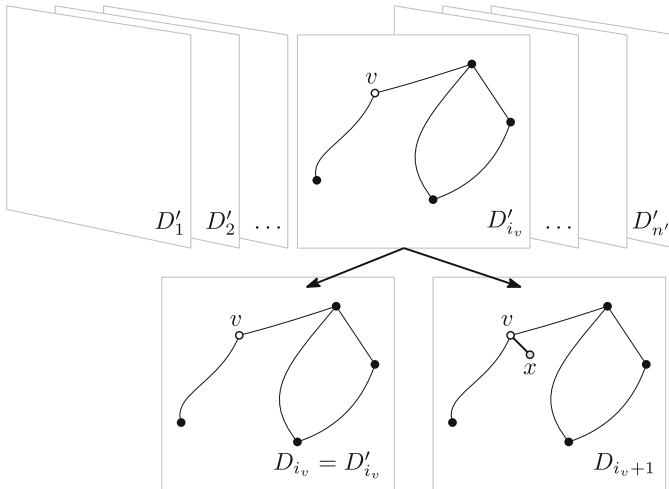


Fig. 6. Illustration for **Case A** of the proof of Lemma 4.

Based on Lemma 4, we can construct an equivalent instance of G of size $O(2^k)$ and use it to conclude the proof of Theorem 4 (see [4]).

3.3 Parameterization by Feedback Edge Set

A *feedback edge set* of a graph $G = (V, E)$ is a set $F \subseteq E$ whose removal results in an acyclic graph, and the *feedback edge set number* of G is the minimum size of a feedback edge set of G . We prove the following.

Theorem 5. *Let G be a graph with n vertices and feedback edge set of size $\psi = \psi(G)$. Deciding whether G admits a storyplan, and computing one if any, can be done in $O(2^{O(\psi \log \psi)} + n^2)$ time.*

Algorithm Description. A k -chain of G is a path with $k+2$ vertices and such that its k inner vertices all have degree two. We define two reduction rules.

R.A: *If there exists a vertex of degree one, then remove it from G .*

R.B: *If there exists a k -chain with $k \geq 3$, then remove its inner vertices from G .*

Based on the above reduction rules we can prove the following.

Lemma 5 (*). *STORYPLAN parameterized by feedback edge set number admits a kernel of linear size.*

To conclude the proof of Theorem 5, observe that computing a linear kernel G^* of G , i.e., applying exhaustively the reduction rules **R.A** and **R.B**, can be done in $O(n + \psi)$ time. Afterwards, following the lines of the proof of Theorem 4, we can brute-force a solution for G^* (if any) in $2^{O(\psi \log \psi)}$ time, and reinsert the missing $O(n)$ vertices each in $O(n)$ time (as detailed in [4, Lemma 9]).

3.4 Partial 3-trees

A k -tree has a recursive definition: A complete graph with k vertices is a k -tree; for any k -tree H , the graph obtained from H by adding a new vertex v connected to a clique C of H of size k is a k -tree; C is the *parent clique* of v . A *partial k -tree* is a subgraph of a k -tree and partial k -trees are exactly the graphs of treewidth at most k . Since 2-trees are planar, they admit a storyplan by Theorem 1. We prove that the same holds for partial 3-trees (which may be not planar).

Theorem 6 (*). *Every partial 3-tree G with n vertices admits a storyplan, which can be computed in $O(n)$ time.*

Proof (Sketch). We shall assume that G is a (non-partial) 3-tree. Indeed, if G is a partial 3-tree, a supergraph of G that is a 3-tree always exists by definition. We now construct a specific tree decomposition \mathcal{T} of G that will be used to compute its storyplan; refer to Fig. 7. For a definition of tree decomposition see [13]. The subgraph C_μ induced by the vertices of each bag μ of \mathcal{T} is the *subgraph associated with μ* and it is a 4-clique for each bag μ of \mathcal{T} , except for the root ρ of \mathcal{T} for which

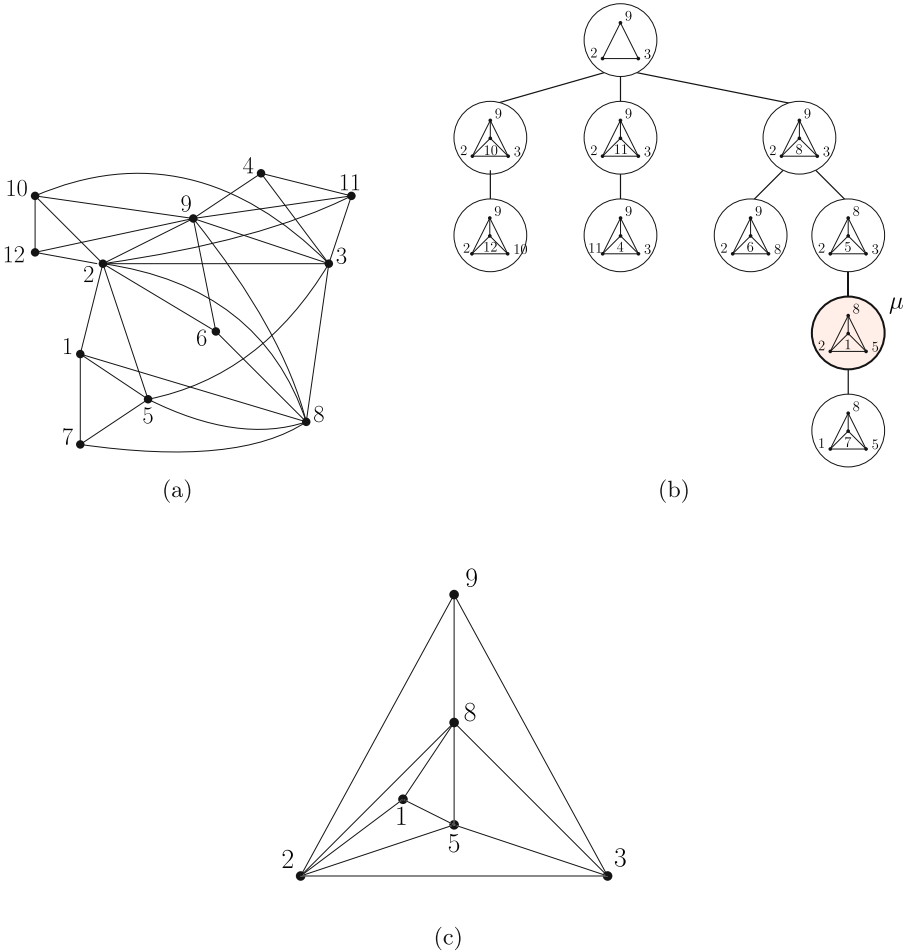


Fig. 7. (a) A 3-tree G . (b) The decomposition tree \mathcal{T} of G . (c) The subgraph G_μ of G associated with bag μ , highlighted in (b).

C_ρ is the initial 3-cycle. The subgraph C_μ contains four 3-cliques, three of them are *active* (this means that they can appear in some subgraph C_ν associated with a child ν of μ) and one is *non-active*. The unique 3-clique in C_ρ is active. Each bag μ of \mathcal{T} has one child ν for each vertex v whose parent clique is an active 3-clique of μ . The 4-clique C_ν consists of the parent clique C of v , vertex v , and the edges connecting v to C ; C is non-active in ν , while the other three 3-cliques are active. For each bag μ distinct from ρ , we denote by v_μ the vertex shared by the three active 3-cliques of C_μ . We say that v_μ is *associated with* μ . One easily verifies that \mathcal{T} is a tree decomposition. Also, \mathcal{T} has $n - 2$ bags: the root and a bag for each vertex of G that is not in the initial 3-cycle.

We now associate to each bag μ a subgraph G_μ of G . For the root ρ , the subgraph G_ρ is the initial 3-cycle. For a bag μ with parent λ , the subgraph G_μ is obtained from G_λ by connecting v_μ to the vertices of its parent clique.

Property 1. Every graph G_μ is an embedded planar 3-tree such that the active 3-cliques of C_μ are internal faces of G_μ .

The proof of Property 1 is by induction on the length of the path from the root ρ to μ in \mathcal{T} . The graph G_ρ consists of a 3-cycle, which is the unique active 3-clique and which is both an internal and an external face. The graph G_μ is obtained by adding the vertex v_μ to G_λ and connecting it to its parent clique C , which is active in C_λ . By induction, C is an internal face of G_λ and therefore, by placing v_μ inside this face, we obtain an embedded planar 3-tree such that the active faces of C_μ are the three faces created by the addition of v_μ inside C .

Let $\mu \neq \rho$ be a bag of \mathcal{T} ; the next property follows from the definition of \mathcal{T} .

Property 2. The neighbors of v_μ distinct from those of its parent clique are all vertices associated with bags of the subtree of \mathcal{T} rooted at μ .

Let $\rho = \mu_1, \mu_2, \dots, \mu_{n-2}$ be an order of the bags of \mathcal{T} according to a preorder visit of \mathcal{T} . To create a storyplan of G , we define an ordering $\tau : v_1, v_2, \dots, v_n$ of the vertices of G such that v_1, v_2 , and v_3 are the vertices of the initial 3-cycle, and each v_i with $i > 3$ is the vertex associated with μ_{i-2} .

Let G_i be the graph induced by the vertices that are visible at step i , for $i \geq 3$; by Property 2 the graph G_i is a subgraph of G_{μ_i} which, by Property 1 is an embedded planar 3-tree such that the three active 3-cliques of C_{μ_i} are faces of G_{μ_i} . To simplify the description we prove that there exists a storyplan $\mathcal{S} = \langle \tau, \{D_i\}_{i \in [n]} \rangle$, where each D_i is a drawing of G_{μ_i} . This implies that there exists a storyplan where each D_i is a drawing of G_i . Let μ_j be the parent of μ_i in \mathcal{T} ; since the order τ corresponds to a preorder of the bags of \mathcal{T} , we have $j < i$. Moreover, all bags μ_k with $j < k < i$, if any, belong to the subtrees of μ_j visited before μ_i and for each such subtree \mathcal{T}' no other bag of \mathcal{T}' exists before μ_j or after μ_i . By Property 2 all the vertices associated with the bags μ_k that belong to $G_{\mu_{i-1}}$ do not have any neighbor after v_{μ_i} and therefore they can be removed. The removal of these vertices transforms $G_{\mu_{i-1}}$ into G_{μ_j} (all the vertices associated with the bags μ_k for $j < k < i$ had been added to G_{μ_j} that had never been changed). By Property 1 the active 3-cliques of G_{μ_j} are faces of G_{μ_j} . It follows that there exists a storyplan $\mathcal{S} = \langle \tau, \{D_i\}_{i \in [n]} \rangle$ whose frames D_i are as follows. D_1 is a planar drawing of a 3-cycle; given D_{i-1} of $G_{\mu_{i-1}}$, a drawing D_i of G_{μ_i} can be computed by removing all vertices associated with the bags μ_k for $j < k < i$, and adding v_{μ_i} inside a face of D_j .

The above storyplan can be computed in $O(n)$ time, see [4].

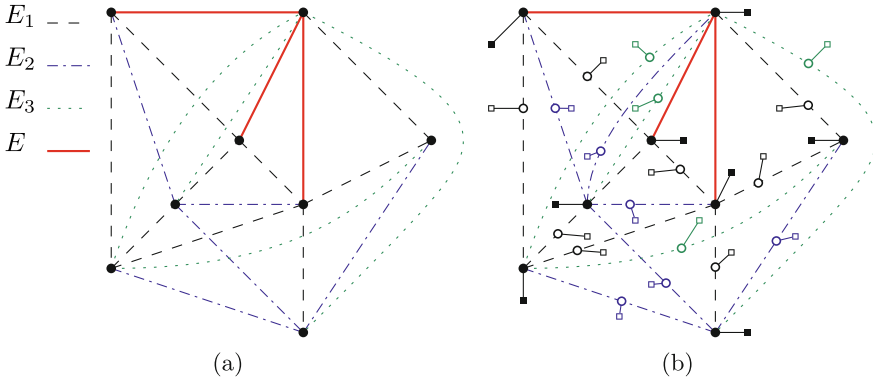


Fig. 8. Illustration for Theorem 7. (a) An instance G_1, G_2, G_3 of SUNFLOWER SEFE; (b) The instance G constructed from G_1, G_2, G_3 ; the subdivision vertices are circles with a white fill while the spectators are squares.

4 Complexity with Fixed Order

In this section we study the STORYPLANFIXEDORDER variant of STORYPLAN, defined below. This variant is closer to the setting studied in [6] and it models the case in which the way the graph changes over time is prescribed.

STORYPLANFIXEDORDER
Input: Graph $G = (V, E)$ with n vertices, total order $\tau : V \rightarrow [n]$
Question: Does G admit a storyplan $\mathcal{S} = \langle \tau, \{D_i\}_{i \in [n]} \rangle$?

We prove that STORYPLANFIXEDORDER is NP-complete by reducing from SUNFLOWER SEFE. Let G_1, \dots, G_k be k graphs on the same set V of vertices. A simultaneous embedding with fixed edges (SEFE) of G_1, \dots, G_k consists of k planar drawings $\Gamma_1, \dots, \Gamma_k$ of G_1, \dots, G_k , respectively, such that each vertex is mapped to the same point in every drawing and each shared edge is represented by the same simple curve in all drawings sharing it. The SEFE problem asks whether k input graphs on the same set of vertices admit a SEFE, and it is NP-complete even when the pairwise intersection between any two input graphs is the same over all pairs of graphs [1, 14]. This variant is called SUNFLOWER SEFE, and the result in [1, 14] proves NP-completeness already when $k = 3$.

Construction. Refer to Fig. 8 for an example. Let G_1, G_2, G_3 be an instance of SUNFLOWER SEFE. Let V be the common vertex set of the three graphs, let E be the common edge set, and let E_i be the exclusive edge set of G_i for $i = 1, 2, 3$. We construct an instance $\langle G, \tau \rangle$ of STORYPLANFIXEDORDER as follows. Graph G contains all vertices in V and all edges in E . Also, for each edge $e = uv$ in E_i , it contains a vertex w_e^i , called a subdivision vertex of E_i , and the edges uw_e^i and vw_e^i (i.e., it contains the edge e subdivided once). Moreover, for each vertex z , either a vertex in V or a subdivision vertex of an edge, G contains an additional

vertex s_z , called the *spectator of z* , and the edge zs_z . To obtain the total order τ we group the vertices of G in a set of blocks B_1, \dots, B_8 , and we order the blocks by increasing index, while vertices within the same block can be ordered arbitrarily. We denote by τ_i^- and τ_i^+ the position in τ of the first and of the last vertex of B_i , respectively, for each $i = 1, \dots, 8$. Block B_1 contains all vertices in V ; for $i \in \{2, 4, 6\}$, block B_i contains all subdivision vertices of $E_{\frac{i}{2}}$, while block B_{i+1} contains all spectators of the vertices in B_i ; finally B_8 contains all spectators of the vertices in B_1 .

Theorem 7 (\star). *The STORYPLANFIXEDORDER problem is NP-complete.*

Proof (Sketch). At a high level, the total order τ is designed to show the three graphs one by one while keeping the common edge set visible. In particular, a spectator vertex s_v forces vertex v to stay visible until s_v appears, while a subdivision vertex w_e^i makes edge e visible only when G_i must be drawn.

5 Discussion and Open Problems

Our work can stimulate further research based on several possible directions.

- It would be interesting to study further parameterizations of STORYPLAN. Is STORYPLAN parameterized by treewidth (pathwidth) in XP? In addition, we note that if the total order is fixed, then an FPT algorithm in the size of the largest frame (or the length of the longest lifespan) readily follows from the proof of Theorem 3.
- Conditions (iii) and (iv) of the definition of a storyplan can be replaced by the existence of a sequence of planar embeddings in which common subgraphs keep the same embedding. This is not true if we study more geometric versions of the problem, in which for instance edges are straight-line segments and/or vertices are restricted on an integer grid of fixed size (as in [6]).
- Condition (ii) of storyplan can be relaxed so to only allow specific crossing patterns [9, 10], e.g., right-angle crossings or few crossings per edge.

Acknowledgement. Research in this work started at the Bertinoro Workshop on Graph Drawing 2022.

References

1. Angelini, P., Da Lozzo, G., Neuwirth, D.: Advancements on SEFE and partitioned book embedding problems. *Theoret. Comput. Sci.* **575**, 71–89 (2015). <https://doi.org/10.1016/j.tcs.2014.11.016>
2. Beck, F., Burch, M., Diehl, S., Weiskopf, D.: The state of the art in visualizing dynamic graphs. In: Borgo, R., Maciejewski, R., Viola, I. (eds.) *EuroVis 2014*. Eurographics Association (2014). <https://doi.org/10.2312/eurovisstar.20141174>
3. Binucci, C., Brandes, U., Di Battista, G., Didimo, W., Gaertler, M., Palladino, P., Patrignani, M., Symvonis, A., Zweig, K.A.: Drawing trees in a streaming model. *Inf. Process. Lett.* **112**(11), 418–422 (2012). <https://doi.org/10.1016/j.ipl.2012.02.011>

4. Binucci, C., Di Giacomo, E., Lenhart, W.J., Liotta, G., Montecchiani, F., Nöllenburg, M., Symvonis, A.: On the complexity of the storyplan problem. CoRR abs/2209.00453 (2022). <https://arxiv.org/abs/2209.00453>
5. Bodlaender, H.L., Kloks, T.: Efficient and constructive algorithms for the path-width and treewidth of graphs. *J. Algorithms* **21**(2), 358–402 (1996). <https://doi.org/10.1006/jagm.1996.0049>
6. Borrazzo, M., Da Lozzo, G., Di Battista, G., Frati, F., Patrignani, M.: Graph stories in small area. *J. Graph Algorithms Appl.* **24**(3), 269–292 (2020). <https://doi.org/10.7155/jgaa.00530>
7. Da Lozzo, G., Rutter, I.: Planarity of streamed graphs. *Theoret. Comput. Sci.* **799**, 1–21 (2019). <https://doi.org/10.1016/j.tcs.2019.09.029>
8. Di Giacomo, E., Didimo, W., Liotta, G., Montecchiani, F., Tollis, I.G.: Techniques for edge stratification of complex graph drawings. *J. Vis. Lang. Comput.* **25**(4), 533–543 (2014). <https://doi.org/10.1016/j.jvlc.2014.05.001>
9. Didimo, W., Liotta, G., Montecchiani, F.: A survey on graph drawing beyond planarity. *ACM Comput. Surv.* **52**(1), 1–37 (2019). <https://doi.org/10.1145/3301281>
10. Hong, S.-H., Tokuyama, T. (eds.): *Beyond Planar Graphs*. Springer, Singapore (2020). <https://doi.org/10.1007/978-981-15-6533-5>
11. Monien, B., Sudborough, I.H.: Min Cut is NP-complete for edge weighted trees. *Theor. Comput. Sci.* **58**, 209–229 (1988). [https://doi.org/10.1016/0304-3975\(88\)90028-X](https://doi.org/10.1016/0304-3975(88)90028-X)
12. Robertson, N., Seymour, P.D.: Graph minors. I. Excluding a forest. *J. Comb. Theory, Ser. B* **35**(1), 39–61 (1983). [https://doi.org/10.1016/0095-8956\(83\)90079-5](https://doi.org/10.1016/0095-8956(83)90079-5)
13. Robertson, N., Seymour, P.: Graph minors. II. Algorithmic aspects of tree-width. *J. Algorithms* **7**(3), 309–322 (1986). [https://doi.org/10.1016/0196-6774\(86\)90023-4](https://doi.org/10.1016/0196-6774(86)90023-4)
14. Schaefer, M.: Toward a theory of planarity: Hanani-Tutte and planarity variants. *J. Graph Algorithms Appl.* **17**(4), 367–440 (2013). <https://doi.org/10.7155/jgaa.00298>
15. Skambath, M., Tantau, T.: Offline drawing of dynamic trees: algorithmics and document integration. In: Hu, Y., Nöllenburg, M. (eds.) *Graph Drawing and Network Visualization (GD'16)*. LNCS, vol. 9801, pp. 572–586. Springer (2016). https://doi.org/10.1007/978-3-319-50106-2_44
16. Vehlou, C., Beck, F., Weiskopf, D.: Visualizing dynamic hierarchies in graph sequences. *IEEE Trans. Vis. Comput. Graph.* **22**(10), 2343–2357 (2016). <https://doi.org/10.1109/TVCG.2015.2507595>