

Outerplanar and Forest Storyplans

Jiří Fiala¹, Oksana Firman², Giuseppe Liotta³, Alexander Wolff², and Johannes Zink²

¹ Charles University, Prague, Czech Republic

² Universität Würzburg, Würzburg, Germany

³ Università degli Studi di Perugia, Perugia, Italy

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 outerplanar storyplan, every frame is outerplanar; in a forest storyplan, every frame is acyclic. We identify graph families that admit such storyplans and families for which such storyplans do not always exist. In the affirmative case, we present efficient algorithms that produce straight-line storyplans.

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]): once a vertex disappears, it may not re-appear. Hence, after a vertex appears, it remains visible until all its incident edges are represented; then it disappears in the transition to the next frame. See Fig. 1 for a storyplan.

Since edge crossings are a natural obstacle to the readability of a graph layout [10], Binucci et al. [4] introduced and studied the *planar storyplan problem* that asks whether a graph G admits a storyplan such that every frame is a crossing-free drawing and in every frame a single new vertex appears. Binucci et al. showed that the problem is NP-complete in general and fixed-parameter tractable w.r.t. the vertex cover 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 let the classes

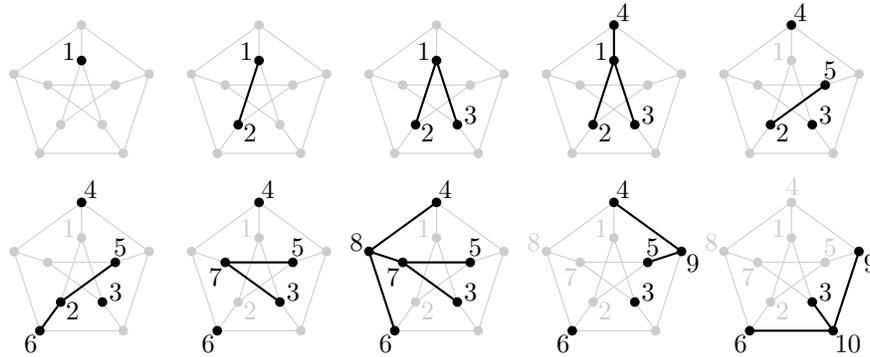


Fig. 1: A forest storyplan of the Petersen graph.

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. We call storyplans with this property *straight-line storyplans*.

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, 3, 6, 7]) and to recent work about graph stories (see, e.g., [5, 9]). The key difference to our work is that these papers (except [4]) assume that the order of the vertices is given as part of the input. We now summarize our contribution, using Δ -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 Fig. 2) by showing that
 - there is a Δ -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
 - there is a Δ -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 is a triangle. We show that no triangulation (except for K_3) admits an outerplanar storyplan; see Section 3.

- We show that every partial 2-tree and every subcubic graph except K_4 admits an outerplanar straight-line storyplan (in linear time); see Section 4. In our construction for subcubic graphs, every frame contains at most five edges.
- A graph must be Δ -free in order to admit a forest storyplan. We show that Δ -free subcubic graphs (as the Petersen graph in Fig. 1), and Δ -free planar graphs admit straight-line forest storyplans (which we can compute in linear and polynomial time, respectively); see Section 5.

We start with some preliminaries in Section 2 and close with open problems in Section 6. We postpone the proofs of statements with a (clickable) “ \star ” to the appendix. Given a positive integer n , we use $[n]$ as shorthand for $\{1, 2, \dots, n\}$.

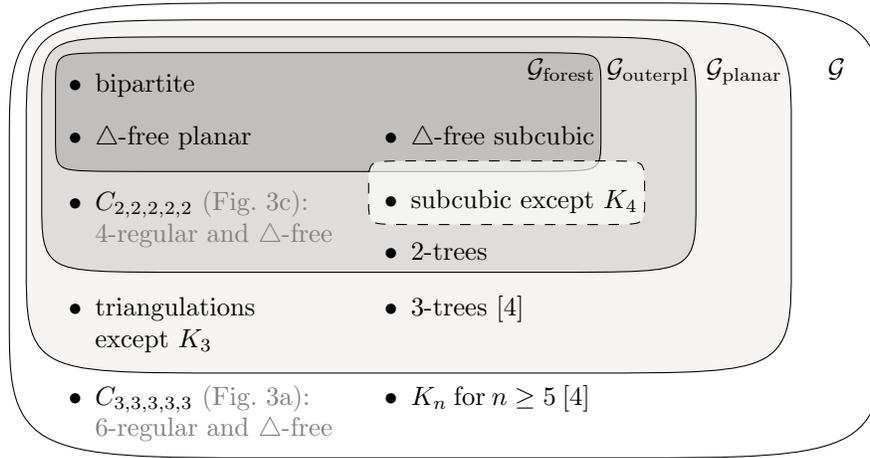


Fig. 2: Overview: existing [4] and new storyplan results, implying $\mathcal{G}_{\text{forest}} \subsetneq \mathcal{G}_{\text{outerpl}} \subsetneq \mathcal{G}_{\text{planar}} \subsetneq \mathcal{G}$. (For simplicity, we mention 2-/3-trees rather than partial 2-/3-trees.)

2 Preliminaries

Our definitions of a planar, an outerplanar, and a forest storyplan are based on the definition of a planar storyplan of Binucci et al. [4].

Definition 1. A **planar 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 interval $[i_v, j_v]$ is the **lifespan** of 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 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 for the definition of a storyplan we allow that edges could be represented by curves, our constructions use only straight-line segments. 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 $\langle D_1, D'_1, \dots, D_{n-1}, D'_{n-1}, D_n \rangle$, 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. Similar to D'_i , we define G'_i for $i \in [n-1]$ as the graph induced by the vertices of $V(G_i) \cap V(G_{i+1})$. We now list some useful observations.

Property 1. 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 1 ([4]). *Let $K_{a,b} = (A \cup B, E)$ be a complete bipartite graph with $a = |A|$, $b = |B|$, and $3 \leq a \leq b$. Let $\mathcal{S} = \langle \tau, \{D_i\}_{i \in [a+b]} \rangle$ be a planar storyplan of $K_{a,b}$. Exactly one of A and B is such that all its vertices are visible at some $i \in [a+b]$.*

Example 1. 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.

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 1. *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. \square

Example 2 (Platonic graphs). According to Theorem 1, 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 1. The dodecahedron is Δ -free and cubic; hence, it admits a forest storyplan due to Theorem 5. For an ordering of the vertices of the dodecahedron that corresponds to a forest storyplan, see Fig. 13 in the appendix.

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

Theorem 2. *The following statements hold:*

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

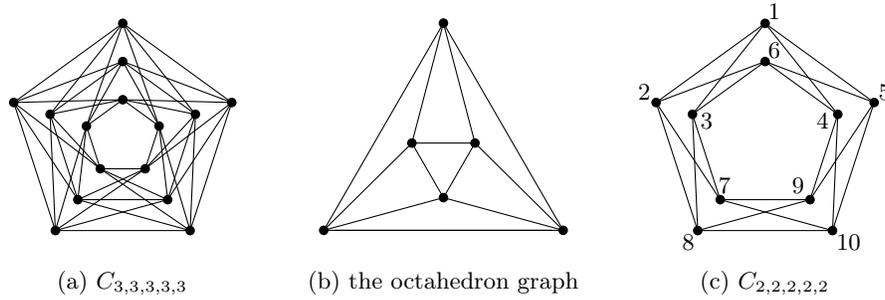


Fig. 3: Three graphs from the proof of Theorem 2. The graph in (a) is Δ -free and does not admit any planar storyplan. The octahedron graph in (b) does not admit any outerplanar storyplan. The graph in (c) is Δ -free and does not admit any forest storyplan (but the vertex numbering corresponds to an outerplanar storyplan – if vertex 8 is placed at the position of vertex 6, which will have disappeared by then).

Proof. 1. The graph $C_{3,3,3,3,3}$ (see Fig. 3a) is Δ -free and 6-regular, but does not admit a planar storyplan as we will now show. Let $V(G) = V_1 \cup \dots \cup V_5$ be the partition of the vertex set into independent sets of size 3. Note that, for $i \in \{1, 2, 3, 4, 5\}$, $G[V_i \cup V_{(i \bmod 5)+1}]$ is isomorphic to $K_{3,3}$. For $K_{3,3} = G[V_1 \cup V_2]$, we know by Lemma 1 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 . Hence, for a frame to be planar, the vertices of V_2 and V_5 cannot be shown simultaneously. This, in turn, means that the vertices of V_3 and V_4 must be shown simultaneously. But then 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 Fig. 3b) is planar, 4-regular, and K_4 -free, but does not admit an outerplanar storyplan due to Example 2.

3. The graph $C_{2,2,2,2,2}$ (see Fig. 3c) is Δ -free and 4-regular, but does not admit a forest storyplan. The proof is analogous to the one above. 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 Fig. 3c yields an outerplanar storyplan. Note that we cannot use the vertex positions exactly as in the figure, but if we place vertex 8 at the position of vertex 6 (which will have disappeared by then), every frame is crossing-free.

□

4 Outerplanar Storyplans

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

Theorem 3. *Every partial 2-tree admits a straight-line outerplanar storyplan, and such a storyplan can be computed in linear time.*

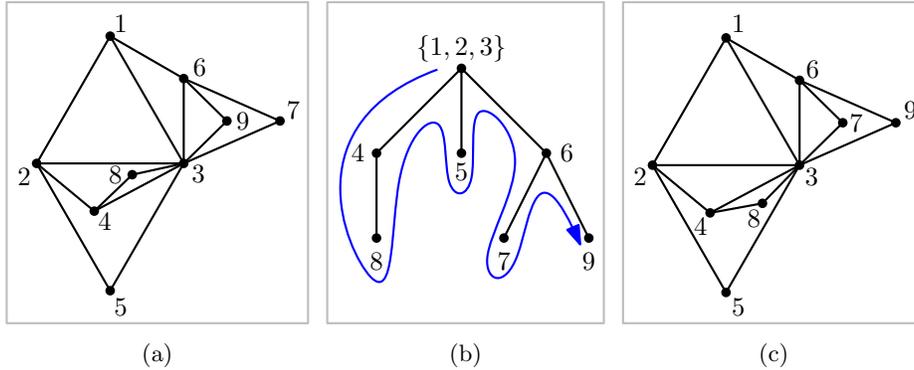


Fig. 4: A 2-tree G with a stacking order (a); its tree decomposition yields a vertex order $\sigma = \langle 1, 2, 3, 4, 8, 5, 6, 7, 9 \rangle$ (b); and an embedding of G that together with σ defines an outerplanar storyplan (c).

Proof. Due to Property 1, it suffices to prove the statement for 2-trees.

Let G be a 2-tree. Hence, there exists a *stacking order* $\sigma = \langle v_1, \dots, v_n \rangle$ of the vertex set $V(G)$. 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 [8, 13]. 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 straight-line storyplan. Fig. 4(a) shows a 2-tree with a stacking order (that is not an outerplanar storyplan).

To show that an outerplanar storyplan always exists, we create a tree decomposition $T_{G, \sigma}$ 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 $i = 4, 5, \dots$, let v_i of σ be stacked onto the edge $v_k v_\ell$ with $k < \ell < i$. We add a node to $T_{G, \sigma}$ that represents v_i and is a child of the node representing v_ℓ . Note that if $\ell \leq 3$, then this new node is a child of the root. Fig. 4(b) shows a tree decomposition of the 2-tree in Fig. 4(a).

From $T_{G, \sigma}$, we obtain a vertex order $\sigma' = \langle v'_1, v'_2, \dots, v'_n \rangle$ being an outerplanar storyplan as follows; see Fig. 4(c). Let $v'_1 = v_1$, $v'_2 = v_2$, and $v'_3 = v_3$. Now, we traverse the nodes of $T_{G, \sigma}$ in (depth-first) pre-order and add the represented vertices of G to σ' . We claim that for σ' , we can choose an embedding \mathcal{E} (defined implicitly next) of G such that all frames are outerplanar. Note that the first three vertices form a triangle, which always admits an outerplanar drawing. Now consider v'_i for $i = 4, 5, \dots$. 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 $k \in \{j + 1, \dots, i - 1\}$, 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. \square

Theorem 4 (\star). *Every subcubic graph except K_4 admits a straight-line outerplanar storyplan with at most five edges in each frame, and such a storyplan can be computed in linear time.*

Proof. Due to Property 1, 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. For an outerplanar storyplan, we will order the vertices v_1, \dots, v_n of G such that the resulting sequence of graphs $\langle G_1, G'_1 \dots, G_{n-1}, G'_{n-1}, G_n \rangle$ has the following property: for $4 \leq i \leq n - 1$, G'_i has at most two edges. Only for $i = 3$, G'_i may be a triangle and would thus contain three edges. Then we show how to obtain outerplanar drawings $D_1, D'_1 \dots, D_{n-1}, D'_{n-1}, D_n$ of the graphs $G_1, G'_1 \dots, G_{n-1}, G'_{n-1}, G_n$, respectively. For $i \in [n]$, let $H_i = G[\{v_1, \dots, v_i\}]$.

We pick the first vertex v_1 arbitrarily. For $1 < i \leq n$, let v denote a vertex of G'_{i-1} with maximum degree in H_{i-1} . If there are more choices, let v additionally have maximum degree in G'_{i-1} . We then select $v_i \in V(G) \setminus \{v_1, \dots, v_{i-1}\}$ as a neighbor of v in G . 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.

We claim that, for $4 \leq i \leq n - 1$, the graph G'_i contains at most two edges. In addition, if G'_i contains two edges, then these edges are both incident with v_i . This would mean that, 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 Fig. 10), then G_4 still has at most five edges since G is not K_4 . Clearly, D_1 and D_2 have at most two edges.

We consider three cases depending on the degree of v in G'_{i-1} ; see Fig. 5.

- (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} because H_{i-1} is connected and, for an edge in G'_{i-1} , H_{i-1} contains an incident degree-2 vertex. Note that all edges in G_i are new and incident with 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 . Hence, G'_i has at most two edges. Note that both edges are incident with 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 construction, all other vertices in G_{i-1} have also 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.

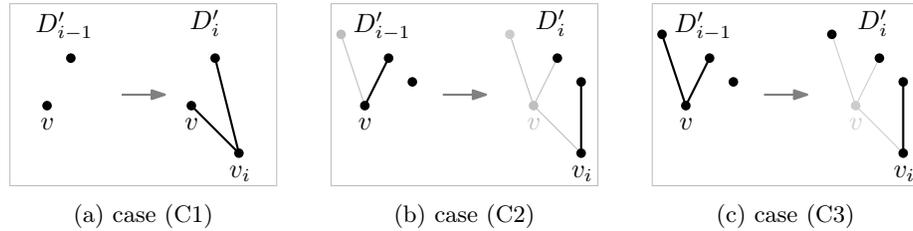


Fig. 5: Cases considered in the proof of Theorem 4. In all of them, the number of edges in G'_i is maximized. Gray vertices and edges were visible in some previous steps.

(C3) Vertex v has two neighbors in G'_{i-1} . In this case, the two edges incident with 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. In the appendix, we show that we can always find a position of the vertices such that each frame is outerplanar and straight-line.

To see the linear runtime, note that we can choose v_i and update H_i in amortized constant time by using a suitable data structure [12]. The other steps of our construction require constant time for each vertex v_i . \square

5 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 5 (\star). *Every \triangle -free subcubic graph admits a straight-line forest storyplan. Such a storyplan can be computed in linear time and has at most five edges per frame.*

Proof sketch. We use the storyplan from the proof of Theorem 4. By construction, we never get a cycle since we consider triangle-free graphs. \square

As a warm-up for our main result, we briefly show the following weaker result.

Observation 1 *Every \triangle -free outerplanar graph admits a straight-line 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 $\sigma = \langle v_1, v_2, \dots, v_n \rangle$ be the circular order of the vertices along the outer face of Γ (which can easily be determined in linear time [11]). We claim that σ yields a forest storyplan of G . (Note that the positions of the vertices in Γ will make this storyplan straight-line.)

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 $F = \langle v_{i_1}, v_{i_2}, \dots, v_{i_k} \rangle$ with $i_1 < i_2 < \dots < i_k$ 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. \square

Now we improve upon the simple result above. Note, however, that we do not guarantee a linear running time any more.

Theorem 6. *Every \triangle -free planar graph admits a straight-line forest storyplan, and such a storyplan can be computed in polynomial 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 position 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, which means that we add it 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.

Let $G_1 = G$ and, for $i \in \{1, 2, \dots\}$, let v_i be the vertex that we pick in iteration i , and let G_{i+1} be the subgraph of G_i that we obtain after removing the vertices (and the edges incident to them) that disappear in iteration i ; see Fig. 7b. The algorithm terminates as soon as G_i is a forest and adds the remaining vertices in arbitrary order to the storyplan under construction. We call vertices and edges incident with the (current) outer face *outer*. The others are *inner*.

We always pick outer vertices. For this reason, only two types of vertices are problematic for avoiding cycles: the endpoints of *chords* (i.e., inner edges incident with 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 Figs. 6 and 7c. For example, the edges e and e' of G_2 in Fig. 7b 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 Fig. 7c), 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 with a common edge of G'_i . Note that H_i is outerplane (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:

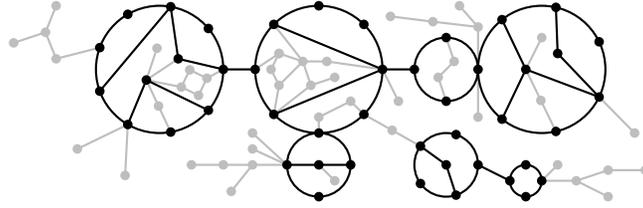


Fig. 6: From an embedded Δ -free planar graph G_i (black & gray), we obtain G'_i (black). Note that G'_i decomposes into seven simple cycles and two connected components. The outer edges and vertices of these connected components form cactus graphs.

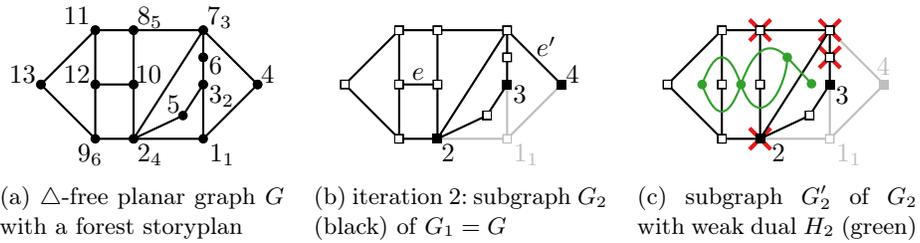


Fig. 7: A Δ -free planar graph G where (a) shows a forest storyplan computed by our algorithm, (b) shows the result of the first iteration of the algorithm, and (c) shows the auxiliary graph for the second iteration. Subscripts refer to the iteration in which a vertex is picked. Red crosses mark vertices that may not be picked.

- (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.
- (I3) *During* iteration i , the only inner edges that may be visible are those that are incident with 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 with 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 Fig. 8. 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 always is one.

- (R1) Do not pick a vertex v whose extended neighborhood $N[v] = \{v\} \cup \{u : uv \in E(G_i)\}$ contains all invisible vertices of the outer face of G'_i .
- (R2) Do not pick an endpoint of a chord.
- (R3) Do not pick a neighbor of an endpoint of a chord if the other endpoint of that chord is visible.
- (R4) Do not pick an endpoint of a half-chord if the other endpoint is visible.

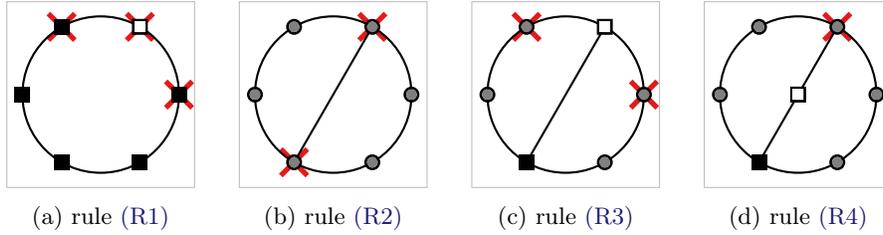


Fig. 8: Rules that determine which vertices may not be picked (marked by red crosses). Black squares represent visible vertices, white squares represent invisible vertices, and gray disks represent vertices that may be visible or invisible.

Rule (R1) ensures that we do not close a cycle on the outer face, thus, invariant (I1) holds. Rule (R4) ensures that none of the visible inner vertices is adjacent to two visible vertices on the outer face (including the picked vertex), thus, invariant (I2) holds. Rules (R2) and (R3) ensure that no chords are visible. Together with rule (R4) and the fact that G is \triangle -free, they ensure that the inner edges that are visible are incident with the picked vertex and no other vertex on the outer face. Thus, invariant (I3) holds. Invariant (I4) holds because we always pick a vertex on the outer face and remove it. As a result, the faces incident with the picked vertex become part of the outer face and the previously inner neighbors (if any) of the picked vertex become incident with 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 (I1), 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 with 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 (I3)). If one endpoint is indeed visible, then its unique neighbor on the boundary of f that is not incident with the chord

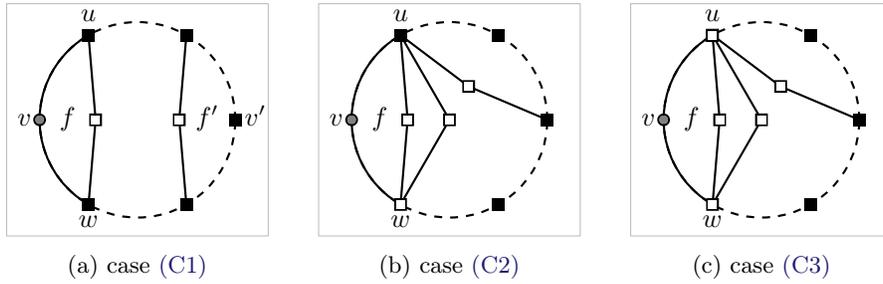


Fig. 9: Cases when there is no chord in G'_i . We always find a good vertex.

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 with 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, which we prove in the appendix.

Claim 1 (\star). The set F has cardinality at least 2.

Claim 2 (\star). Let the edge e (or the edge pair $\{e_1, e_2\}$) be any chord (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 $F_2 \cap F \neq \emptyset$.

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 (R2) and (R3). Let $f \in F$, and let u and w be the endpoints of the unique half-chord incident with 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 (R1) and (R4) are observed by the definition of v . If, 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 Fig. 9. 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 1. 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. W.l.o.g., 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 (R1). If there was another half-chord incident with u , either it would again be incident with w , which does not violate rule (R4), or it would be incident with another vertex of G'_i , which is visible. By Claim 2, 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 (C1) instead.

- (C3) Both u and w are invisible. We claim that u observes the rules. Similar to case (C2), u observes rule (R1) (since w stays invisible) and rule (R4) (if there was another half-chord incident with u whose other endpoint is visible, we would be in case (C1) or in case (C2)).

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 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 facets 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 . We provide the details in the appendix.

Claim 3 (\star). There is a good vertex on the outer face of \hat{G}'_i .

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.

Concerning the running time, note that, if we maintain the outer face, we can find, for each vertex, its incident chords and half-chords in linear time. Further, our constructive proof can be turned into a polynomial-time algorithm as it includes only graph traversal and graph construction operations that can be executed in polynomial time. \square

6 Open Problems

1. What is the complexity of deciding whether a given graph admits an outerplanar or a forest storyplan? We conjecture that recognition is NP-hard.
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.

References

1. M. Abdelaal, A. Lhuillier, M. Hlawatsch, and D. Weiskopf. Time-aligned edge plots for dynamic graph visualization. In E. Banissi et al., editor, *Proc. 24th Int. Conf. Inform. Vis. (IV'20)*, pages 248–257, 2020. doi:10.1109/IV51561.2020.00048.
2. K. Been, M. Nöllenburg, S.-H. Poon, and A. Wolff. Optimizing active ranges for consistent dynamic map labeling. *Comput. Geom.*, 43(3):312–328, 2010. doi:10.1016/j.comgeo.2009.03.006.
3. C. Binucci, U. Brandes, G. Di Battista, W. Didimo, M. Gaertler, P. Palladino, M. Patrignani, A. Symvonis, and K. A. Zweig. Drawing trees in a streaming model. *Inf. Process. Lett.*, 112(11):418–422, 2012. doi:10.1016/j.ipl.2012.02.011.
4. C. Binucci, E. Di Giacomo, W. J. Lenhart, G. Liotta, F. Montecchiani, M. Nöllenburg, and A. Symvonis. On the complexity of the storyplan problem. In P. Angelini and R. von Hanxleden, editors, *GD 2022*, volume 13764 of *LNCS*, pages 304–318. Springer, 2023. URL: <https://arxiv.org/abs/2209.00453>.
5. M. Borrazzo, G. Da Lozzo, G. Di Battista, F. Frati, and M. Patrignani. Graph stories in small area. *J. Graph Algorithms Appl.*, 24(3):269–292, 2020. doi:10.7155/jgaa.00530.
6. M. 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.
7. G. Da Lozzo and I. Rutter. Planarity of streamed graphs. *Theor. Comput. Sci.*, 799:1–21, 2019. doi:10.1016/j.tcs.2019.09.029.
8. H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combin.*, 10(1):41–51, 1990. doi:10.1007/BF02122694.
9. G. Di Battista, W. Didimo, L. Grilli, F. Grosso, G. Ortali, M. Patrignani, and A. Tappini. Small point-sets supporting graph stories. In P. Angelini and R. von Hanxleden, editors, *GD 2022*, volume 13764 of *LNCS*, pages 289–303. Springer, 2023. URL: <https://arxiv.org/abs/2208.14126>.
10. G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, 1999.
11. G. Di Battista and F. Frati. Small area drawings of outerplanar graphs. *Algorithmica*, 54(1):25–53, 2009. doi:10.1007/11618058_9.
12. D. W. Matula and L. L. Beck. Smallest-last ordering and clustering and graph coloring algorithms. *J. ACM*, 30(3):417–427, 1983. doi:10.1145/2402.322385.
13. W. Schnyder. Embedding planar graphs on the grid. In *Proc. 1st ACM-SIAM Symp. Discrete Algorithms (SODA'90)*, pages 138–148, 1990. URL: <https://dl.acm.org/doi/10.5555/320176.320191>.

Appendix

Theorem 4 (\star). *Every subcubic graph except K_4 admits a straight-line outerplanar storyplan with at most five edges in each frame, and such a storyplan can be computed in linear time.*

Proof. The first part of the proof is in the main part of the paper. Now we show that we can always find a position of the vertices such that the drawings $D_1, D'_1, \dots, D'_{n-1}, D_n$ of the graphs $G_1, G'_1, \dots, G'_{n-1}, G_n$, respectively, are outerplanar and straight-line.

It is obvious that the drawings $D_1, D'_1, D_2, D'_2, D_3, D'_3$ are outerplanar even if they are straight-line. Now we want to show, for $i \in \{4, \dots, n\}$, how to place v_i such that D_i is outerplanar and straight-line. Let v_i be 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 vv' and vv'' 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 Δuvv_i and/or the triangle Δvvv_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 Δuvv_i and Δvvv_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 from v that contain v' and v'' , respectively. There are three cases: Δuvw 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 union (in the first case) or in the intersection (in the second case) of the open halfplanes bounded by the lines through vu and vw that contain neither v' nor v'' , see Fig. 11a and Fig. 11b. Now assume that Δuvw 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' .

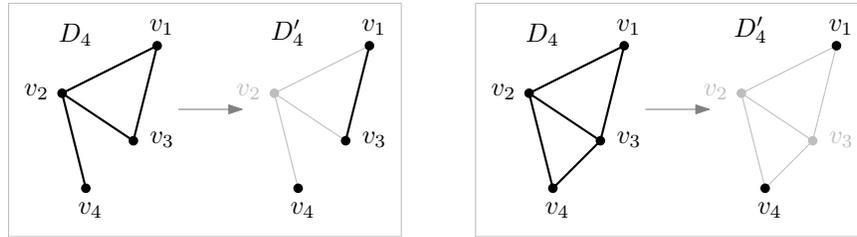
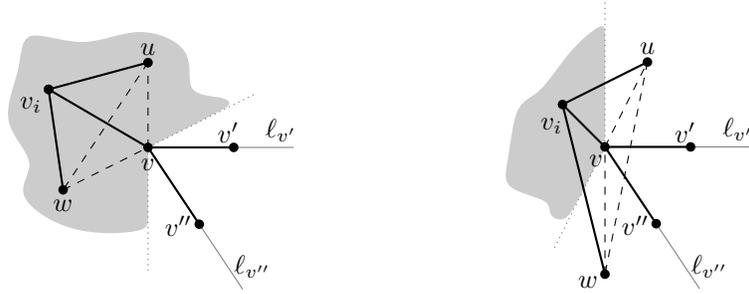
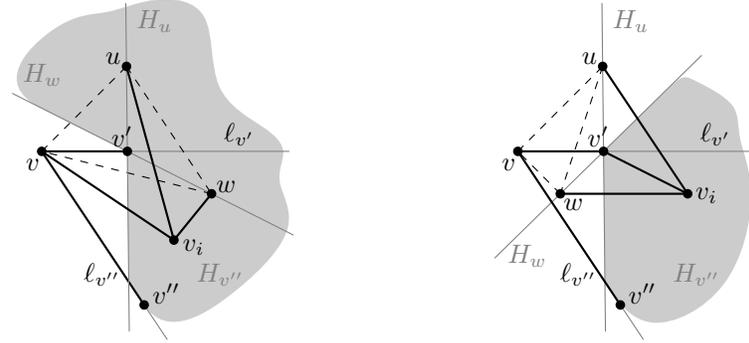


Fig. 10: Illustration of the special case that G'_3 contains three edges. Then v_4 has one (left) or two (right) neighbors in H_3 , but G'_4 has at most one edge. Grey vertices and edges are not part of the graphs, but were visible in the previous step.



(a) vuw intersects neither $l_{v'}$ nor $l_{v''}$. (b) vuw intersects both $l_{v'}$ and $l_{v''}$.



(c) vuw intersects only $l_{v'}$, uw does not intersect vv' . (d) vuw intersects only $l_{v'}$, uw intersects vv' .

Fig. 11: Various cases of placing v_i to get a straight-line outerplanar drawing D_i .

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 $l_{v''}$ that contains v' . If uw does not intersect vv' , we can place v_i in $((H_u \cup H_w) \cap H_{v''}) \setminus l_{v'}$, see Fig. 11c. If uw intersects vv' , we place v_i in $(H_u \cap H_w \cap H_{v''}) \setminus l_{v'}$, see Fig. 11d. Note that we always can find a position for v_i such that none of the new edges contains a vertex visible in D_i . Hence, we have shown that there is a position for v_i such that the drawing D_i is straight-line and outerplanar. \square

Theorem 5 (\star). *Every Δ -free subcubic graph admits a straight-line forest storyplan. Such a storyplan can be computed in linear time and has at most five edges per frame.*

Proof. Due to Property 1, it suffices to prove the statement for Δ -free cubic graphs. Recall the proof of Theorem 4. 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 Δ -free, 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. \square

Claim 1 (\star). 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 with 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, $|F| \geq 2$. \square

Claim 2 (\star). Let the edge e (or the edge pair $\{e_1, e_2\}$) be any chord (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 $F_2 \cap F \neq \emptyset$.

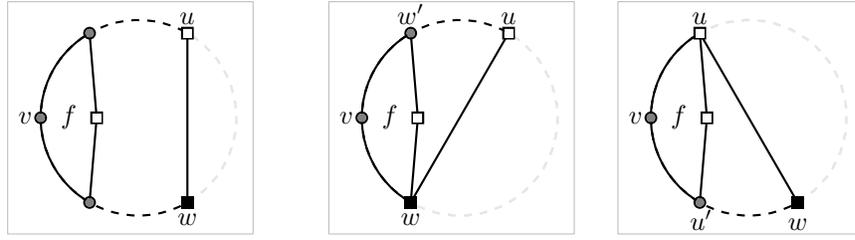
Proof. If we have a chord e , then the dual edge of e is an edge of H_i , which corresponds to a block node of the block-cut tree, which cannot be a leaf (because all leaves represent cycles). Then, however, if we traverse the block-cut tree, in the one or the other direction of the edge, we will find a leaf in both parts and each leaf contains a vertex in F as shown in the proof of Claim 1.

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. \square

Claim 3 (\star). There is a good vertex on the outer face of \hat{G}'_i .

Proof. Let \hat{F} be the subset of F that is contained in \hat{G}'_i , which is non-empty by Claim 2. If there is a face $f \in \hat{F}$ neither incident with u nor w , then we can pick a free vertex v of f ; see Fig. 12a. Since at least u stays invisible, v observes rule (R1). 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 with both u and w as this would create a triangle. Assume that there is a face $f \in \hat{F}$ incident with a half-chord with endpoints w and w' ; see Fig. 12b. We claim that we can pick a free vertex v of f . Since at least u stays invisible, v observes rule (R1). Rules (R2) and (R4) are trivially observed. Rule (R3) is also observed, since w' is not incident to a chord whose



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

Fig. 12: Cases when there are chords in G'_i . We always find a good vertex.

other endpoint is visible (as they are all inside \hat{G}'_i) and w is not incident with a chord whose other endpoint is visible, otherwise invariant (I3) would be violated.

Finally, all faces in \hat{F} are incident with u . For a face $f \in \hat{F}$, let u' be the other endpoint of the half-chord; see Fig. 12c. Note that it might happen now that all free vertices of f break rule (R1) or rule (R3). We claim, however, that we can always pick u' . Because of u , u' observes rule (R1). Furthermore, u' observes rule (R2) and rule (R4) as u' cannot be incident with a chord or a half-chord. If u' was incident with 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 with u due to Claim 2. By the same argument, all chords and half-chords of \hat{G}'_i are incident with u . In order to show that u' observes rule (R3), note that the only vertex of \hat{G}'_i that is incident with 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 (R3) as well and, hence, is a good vertex. \square

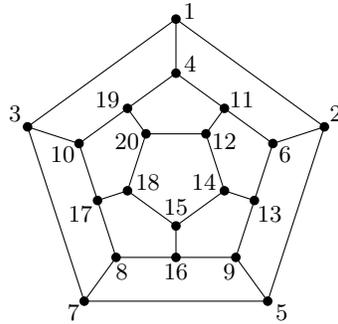


Fig. 13: The dodecahedron with a vertex numbering that corresponds to a forest storyplan.