European Journal of Combinatorics 119 (2024) 103806

Contents lists available at ScienceDirect

European Journal of Combinatorics

journal homepage: https://www.sciencedirect.com/journal/european-journal-of-combinatorics

Testing the planar straight-line realizability of 2-trees with prescribed edge lengths☆

Carlos Alegría, Manuel Borrazzo, Giordano Da Lozzo, Giuseppe Di Battista, Fabrizio Frati, Maurizio Patrignani

Roma Tre University, Rome, Italy

ARTICLE INFO

Article history:
Available online 30 September 2023

ABSTRACT

We study a classic problem introduced thirty years ago by Eades and Wormald. Let $G = (V, E, \lambda)$ be a weighted planar graph, where $\lambda : E \to \mathbb{R}^+$ is a *length function*. The Fixed Edge-Length Planar Realization problem (FEPR for short) asks whether there exists a *planar straight-line realization* of G, i.e., a planar straight-line drawing of G where the Euclidean length of each edge $e \in E$ is $\lambda(e)$.

Cabello, Demaine, and Rote showed that the FEPR problem is NP-hard, even when λ assigns the same value to all the edges and the graph is triconnected. Since the existence of large triconnected minors is crucial to the known NP-hardness proofs, in this paper we investigate the computational complexity of the FEPR problem for weighted 2-trees, which are K_4-minor free. We show the NP-hardness of the problem, even when λ assigns to the edges only up to four distinct lengths. Conversely, we show that the FEPR problem is linear-time solvable when λ assigns to the edges up to two distinct lengths, or when the input has a prescribed embedding. Furthermore, we consider the FEPR problem for weighted maximal outerplanar graphs and prove it to be linear-time solvable if their dual tree is a path, and cubic-time solvable if their dual tree is a caterpillar. Finally, we prove that the FEPR problem for weighted 2-trees is slice-wise polynomial in the length of the large path.

☆ This research was partially supported by MIUR Project "AHeAD" under PRIN 20174LF3T8. A preliminary version of this paper was presented at the 29th International Symposium on Graph Drawing and Network Visualization (GD2021).

E-mail addresses: carlos.alegria@uniroma3.it (C. Alegría), manuel.borrazzo@uniroma3.it (M. Borrazzo), giordano.dalozzo@uniroma3.it (G. Da Lozzo), giuseppe.dibattista@uniroma3.it (G. Di Battista), fabrizio.frati@uniroma3.it (F. Frati), maurizio.patrignani@uniroma3.it (M. Patrignani).

https://doi.org/10.1016/j.ejc.2023.103806

1. Introduction

The problem of producing drawings of graphs with geometric constraints is a core subject in Graph Drawing (see, e.g., [3–5,10,16,21,29,30,40,46,50,52]). In this context, a classic question is the one of testing whether a graph can be drawn planarly so that its edges are straight-line segments of prescribed lengths. The study of such a question has connections with several topics in computational geometry [24,48,55], rigidity theory [23,34,36], structural analysis of molecules [13,35], and sensor networks [19,44,47]. Formally, given a weighted planar graph $G = (V, E, \lambda)$, i.e., a planar graph with vertex set V and edge set E equipped with a *length function* $\lambda : E \to \mathbb{R}^+$, the Fixed Edge-Length Planar Realization problem (FEPR, for short) asks whether there exists a *planar straight-line realization* of G, i.e., a planar straight-line drawing of G in the plane where the Euclidean length of each edge $e \in E$ is $\lambda(e)$. The FEPR problem was first studied by Eades and Wormald [28], who showed its NP-hardness even for triconnected planar graphs and for biconnected planar graphs with unit lengths. Cabello, Demaine, and Rote strengthened this result by proving NP-hardness even for triconnected planar graphs with unit lengths [18]. The graphs that admit a planar straight-line realization with unit lengths are also called *matchstick graphs*. Abel et al. [1] showed that recognizing matchstick graphs is strongly $\exists\mathbb{R}$-complete, solving an open problem stated by Schaefer [49], who proved analogous results for non-planar realizations with unit edge lengths.

Since the known hardness proofs of the FEPR problem are all reductions based on gadgets containing large triconnected minors, we study its computational complexity for weighted 2-trees, which are the maximal graphs with no K_4-minor. A 2-*tree* is a graph composed of 3-cycles glued together along edges in a tree-like fashion. As an example, Fig. 1 shows a planar and a non-planar straight-line realization of the same weighted 2-tree. The class of 2-trees has been deeply studied in Graph Drawing (e.g., in [31,32,38,39,45]); such a graph class coincides with the one of the maximal series–parallel graphs. In particular, the edge lengths of planar straight-line drawings of 2-trees have been studied in [14,15].

Related to our research are the notions of *distance number* and *degenerate distance number* of graphs introduced by Carmi et al. [20]. Let G be a graph. The *distance number* of G is the minimum number of distinct edge lengths in a straight-line drawing of G, where vertices are mapped to distinct points of the plane and no vertex intersects any non-incident edge, however distinct edges are allowed to cross. The *degenerate distance number* of G relaxes the requirements of the distance number, by allowing vertices to intersect non-incident edges. Exploiting polynomial embeddings, Alon and Feldheim have shown that the degenerate distance number of outerplanar graphs is at most 3 [3]. This bound was subsequently improved to 2, which is optimal, by Chol-Un Kim and Kwang-Il Sol [37]. More recently, Bakhajian and Feldheim have shown that the distance number of outerplanar graphs is at most thirteen [7].

First, we show that, in the fixed embedding setting, the FEPR problem can be solved in linear time for 2-trees (Section 3). We remark that the FEPR problem is NP-hard for general weighted planar graphs with fixed embedding [18,28]. Second, we show that, in the variable embedding setting, the FEPR problem is NP-hard for 2-trees when the number of distinct lengths is at least four (Section 4), whereas it is linear-time solvable for 2-trees when the number of distinct lengths is 1 or 2 (Section 5). Note that, for general weighted planar graphs, the problem is NP-hard even when all the edges are required to have the same length [28]. Third, we move our attention to maximal outerplanar graphs (Section 6), which form a notable subclass of 2-trees. We show that the FEPR problem can be solved in linear time for maximal outerpaths, i.e., the maximal outerplanar graphs whose dual tree is a path, and in cubic time for maximal outerpillars, i.e., the maximal outerplanar graphs whose dual tree is a caterpillar. Finally, we present a slice-wise polynomial algorithm for weighted 2-trees, parameterized by the length of the longest path (Section 7). A slice-wise polynomial algorithm with respect to a parameter k is an algorithm with running time $O(n^{f(k)})$, where n is the input size (in our case, the number of vertices of the 2-tree) and f is some

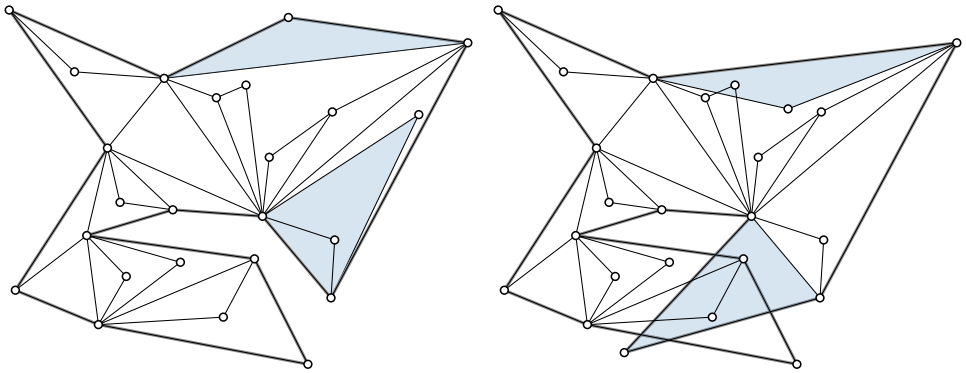

Fig. 1. A planar and a non-planar straight-line realization of the same weighted 2-tree.

computable function (in our case, an exponential function). Section 2 contains preliminaries and Section 8 provides conclusions and open problems.

Similarly to [18], in our algorithms, we adopt the real RAM model of computation, which is customary in computational geometry and allows us to perform standard arithmetic operations on real numbers in constant time. Furthermore, we show NP-hardness in the Turing machine model by exploiting lengths whose encoding has constant size.

2. Preliminaries

Throughout the paper, we assume that graphs are connected and simple (i.e., with no loops or parallel edges). Moreover, whenever we refer to the length of a segment or to the distance of two geometric objects, we always assume that these are measured according to the Euclidean metric.

Drawings and embeddings. A *drawing* Γ of a graph is a mapping of each vertex v to a distinct point $\Gamma(v)$ in the plane and of each edge (u, v) to a Jordan arc $\Gamma(u, v)$ with endpoints $\Gamma(u)$ and $\Gamma(v)$. We often use the same notation for a vertex v and the point $\Gamma(v)$. A drawing is *planar* if it contains no two crossing edges and it is *straight-line* if each curve representing an edge is a straight-line segment. A planar drawing partitions the plane into connected regions, called *faces*. The bounded faces are the *internal faces*, while the unbounded face is the *outer face*. The *boundary* of a face is the circular list of vertices and edges encountered when traversing the geometric border of the face. A graph G is *planar* if it admits a planar drawing. A planar drawing of G defines a clockwise order of the edges incident to each vertex of G; the set of such orders for all the vertices is a *rotation system* for G. Two planar drawings of G are *equivalent* if (i) they define the same rotation system for G and (ii) their outer faces have the same boundaries. An equivalence class of planar drawings is a *plane embedding* (or simply an *embedding*). When referring to a planar drawing Γ of a graph that has a prescribed embedding $\mathcal{E}$, we always imply that Γ is in the equivalence class $\mathcal{E}$; when we want to stress this fact, we say that Γ *respects* $\mathcal{E}$. All the planar drawings respecting the same embedding have the same set of face boundaries. Hence, we can talk about the face boundaries of a graph with a prescribed embedding. With a little overload of terminology, we sometimes write *face* of a graph with a prescribed embedding, while referring to the boundary of the face. A graph is *biconnected* if the removal of any vertex leaves the graph connected. In any planar drawing of a biconnected graph every face is bounded by a simple cycle. Let Γ be a drawing of a graph G and let G' be a subgraph of G; the *restriction* of Γ to G' is the drawing of G' obtained by removing from Γ the vertices and edges of G that are not in G'.

2-Trees and maximal outerplanar graphs. A 2-*tree* is recursively defined as follows. A cycle formed by k edges is a k-*cycle*. A 3-cycle is a 2-tree. Given a 2-tree G containing the edge (u, w), the graph obtained by adding to G a vertex v and two edges (v, u) and (v, w) is a 2-tree. Note that this

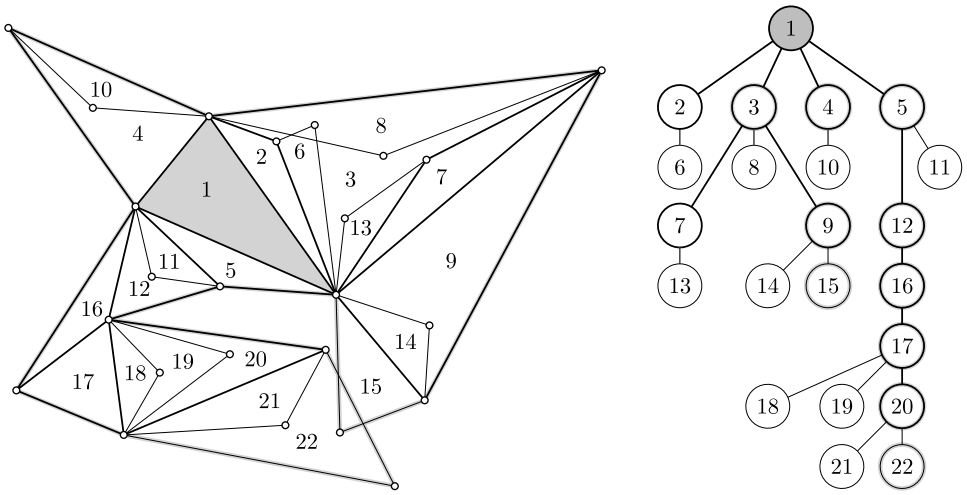

Fig. 2. On the left, a 2-tree G. On the right, the decomposition tree of G rooted at $\triangle_1$. Internal nodes are drawn with thick lines and leaf nodes of are drawn with thin lines.

operation is a 2-clique-sum G and the 3-cycle (u, v, w). A *k-clique-sum* of two graphs H_1 and H_2 is an operation that considers a k-clique K_1 in H_1 and a k-clique K_2 in H_2, then identifies the vertex set of K_1 with the one of K_2, and finally deletes some (possibly all, possibly none) of the edges of such cliques.

We observe that any 2-tree G satisfies the following properties:

(P1) G is biconnected;
(P2) the two neighbors of any degree-2 vertex of G are adjacent; and
(P3) if $|V(G)| \geq 4$, then G contains two non-adjacent degree-2 vertices.

A 2-tree G is essentially composed of 3-cycles glued together along edges in a tree-like fashion. We encode this tree-like structure in a tree we call the *decomposition tree* of G. This tree has a node for each 3-cycle of G, and an edge between two nodes if the corresponding 3-cycles have a common edge[1]; see Fig. 2. Let T be the decomposition tree of G, and n be the number of nodes of G. We observe that T is unique and contains $n - 2$ nodes. Typically, T is *rooted at* a 3-cycle c of G. We can compute T rooted at c by a recursive construction in which, together with T, we produce an auxiliary labeling of the edges of G. An edge e is labeled with the highest node in T whose corresponding 3-cycle contains e. The construction is as follows. If G coincides with c, then T is a unique root node representing c, and each edge of c is labeled with c itself. Otherwise, by Property (P3), there is at least one degree-two vertex not belonging to c. Let v be any such a vertex, let u and w be the vertices adjacent to v, and let c^* be the 3-cycle with vertices u, v, and w. Let T^* denote the decomposition tree of $G - v$ rooted at c. We obtain T by adding to T^* a leaf representing c^*, whose parent is the node corresponding to the label of the edge (u, w). Further, we set to c^* the labels of the edges (v, u) and (v, w). It is not hard to see that this procedure constructs T in $O(n)$ time.

An *outerplanar drawing* is a planar drawing in which all the vertices are incident to the outer face. A graph is *outerplanar* if it admits an outerplanar drawing. An *outerplane embedding* is an equivalence class of outerplanar drawings of a graph; note that a biconnected outerplanar graph has a unique outerplane embedding [43,51]. An outerplanar graph is *maximal* if no edge can be added

[1] We remark that our concept of decomposition tree coincides with a special case of the well-known decomposition of graphs (not necessarily 2-trees) that is commonly called *tree-decomposition*; see, e.g., [27].

to it without losing outerplanarity. In the unique outerplane embedding of a maximal outerplanar graph, every internal face is delimited by a 3-cycle. The *dual tree* T of a biconnected outerplanar graph G is defined as follows. Consider the (unique) outerplane embedding $\mathcal{O}$ of G. Then T has a node for each internal face of $\mathcal{O}$ and has an edge between two nodes if the corresponding faces of $\mathcal{O}$ are incident to the same edge of G. An *outerpath* is a biconnected outerplanar graph whose dual tree is a path. A *caterpillar* is a tree that becomes a path if its leaves are removed. Such path is called the *spine* of the caterpillar. An *outerpillar* is a biconnected outerplanar graph whose dual tree is a caterpillar.

Straight-line realizations. A *weighted graph* $G = (V, E, \lambda)$ is a graph equipped with a length function λ that assigns a positive real number to each edge. A *straight-line realization* of G is a straight-line drawing of G in which each edge e is drawn as a line segment of length $\lambda(e)$. If the drawing is planar, we say that the realization is planar. The Fixed Edge-Length Planar Realization (FEPR) problem receives as input a weighted planar graph G, and asks if there exists a planar straight-line realization of G.

In a straight-line realization of G, each 3-cycle is realized as a triangle. Hence, if the lengths of the edges of at least one 3-cycle do not respect the triangle inequality, then G does not admit any (even non-planar) straight-line realization. We can trivially test in $O(n)$ time whether the length function of an n-vertex weighted 2-tree is such that every 3-cycle satisfies the triangle inequality. Hereafter we assume that every weighted 2-tree satisfies this necessary condition.

In our proofs and algorithms, when clear from the context, we refer interchangeably to the 3-cycles of a 2-tree G, the nodes of its decomposition tree, and the triangles in a straight-line realization of G.

Triangles in a planar straight-line realization. Let $G = (V, E, \lambda)$ be a weighted 2-tree and Γ be a planar straight-line realization of G. We denote with $\triangle_i$ a triangle *drawn* in Γ, that is, the straight-line realization of a 3-cycle of G. Consider two triangles $\triangle_i$ and $\triangle_j$. We say that $\triangle_i$ is *drawn inside* $\triangle_j$ if all the points of $\triangle_i$ are points of $\triangle_j$ and at least one vertex of $\triangle_i$ is an interior point of $\triangle_j$. In the planar straight-line realization shown in Fig. 2, the triangles $\triangle_6$, $\triangle_7$, and $\triangle_8$ are drawn inside the triangle $\triangle_3$. Observe that, if $\triangle_i$ is drawn inside $\triangle_j$, then $\triangle_i$ and $\triangle_j$ have at most two common vertices. On the other hand, in general, $\triangle_i$ can be drawn inside $\triangle_j$ regardless of their distance in the decomposition tree of G. We have the following observations, which we use often (sometimes implicitly) in the paper.

Observation 1. *Let ℓ_i and ℓ_j be the lengths of the longest sides of two triangles $\triangle_i$ and $\triangle_j$, respectively. If $\ell_i > \ell_j$, then $\triangle_i$ is not drawn in $\triangle_j$. If $\ell_i = \ell_j$ and $\triangle_i$ is drawn inside $\triangle_j$, then $\triangle_i$ and $\triangle_j$ share a side of length $\ell_i = \ell_j$ and $\triangle_i$ does not have any other side of length ℓ_i.*

Observation 2. *If $\triangle_i$ is drawn inside $\triangle_j$, then the area of $\triangle_i$ is strictly smaller than the area of $\triangle_j$.*

Of special interest for us is the case in which $\triangle_i$ and $\triangle_j$ share an edge. If such case occurs, then $\triangle_i$ and $\triangle_j$ are either adjacent or siblings in the decomposition tree of G. We have the following.

Observation 3. *Let $\triangle_i$ and $\triangle_j$ be two triangles sharing an edge with end-vertices a and b. Let α_i and β_i (α_j and β_j) denote the interior angles of $\triangle_i$ (resp., $\triangle_j$) at a and b, respectively. If $\triangle_i$ is drawn inside $\triangle_j$, then $\alpha_i < \alpha_j$ and $\beta_i < \beta_j$.*

We conclude the section by defining two specific types of triangles which we are going to use repeatedly in our paper. Consider an isosceles triangle with base of length ℓ_1 and two sides of length ℓ_2, with $\ell_1 \neq \ell_2$. We say the triangle is *tall isosceles* if $\ell_2 > \ell_1$, and *flat isosceles* if instead $\ell_2 < \ell_1$.

Consider now three triangles $\triangle_i$, $\triangle_j$, $\triangle_k$ such that $\triangle_j$ and $\triangle_k$ share a side with $\triangle_i$. If the three triangles have a common side, then a necessary condition for $\triangle_j$ and $\triangle_k$ to be drawn inside $\triangle_i$ is given by Observation 3. If instead there is no common side between $\triangle_j$ and $\triangle_k$, then we have the following lemmas.

Lemma 1. *Let ℓ_1 and ℓ_2 be two lengths such that $2\ell_2 > \ell_1 > \ell_2$, and $\triangle_i$, $\triangle_j$, $\triangle_k$ be three triangles such that:*

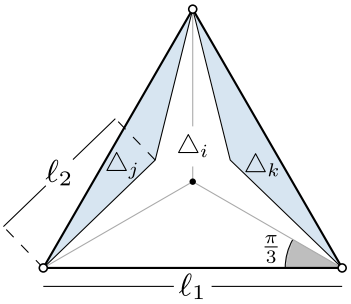

Fig. 3. Illustration for Lemma 1: If two flat isosceles triangles ($\triangle_j$ and $\triangle_k$) are drawn inside an equilateral triangle ($\triangle_i$), then $\frac{\ell_1}{\ell_2} > \sqrt{3}$. The black point is the circumcenter of $\triangle_i$.

- $\triangle_i$ *is an equilateral triangle with sides of length* ℓ_1,
- $\triangle_j$ *and* $\triangle_k$ *are congruent flat isosceles triangles with base of length* ℓ_1 *and two sides of length* ℓ_2,
- $\triangle_i$ *shares one side with* $\triangle_j$ *and another side with* $\triangle_k$, *and*
- $\triangle_j$ *and* $\triangle_k$ *are drawn inside* $\triangle_i$ *and do not overlap with each other.*

Then $\frac{\ell_1}{\ell_2} > \sqrt{3} \simeq 1.73$.

Proof. Refer to Fig. 3. Let v be the common vertex between $\triangle_i$, $\triangle_j$, and $\triangle_k$. Since $\triangle_i$ is equilateral the interior angle of $\triangle_i$ at v is 60°. Hence, since $\triangle_j$ and $\triangle_k$ are congruent and intersect only at v (they do not overlap), the internal angle α of both $\triangle_j$ and $\triangle_k$ at v is at most 30°. We thus have that $\ell_1 = 2\ell_2 \cos(\alpha) > 2\ell_2 \cos(30°) = \sqrt{3}\ell_2$. □

3. Prescribed embedding

In this section, we describe an $O(n)$-time algorithm to solve the FEPR problem for an n-vertex 2-tree whose embedding or whose rotation system is prescribed.

We start by showing how to check in linear time whether a straight-line realization of a 2-tree with prescribed embedding is a planar straight-line realization respecting the embedding. This basic tool will be exploited throughout the paper.

Theorem 4. *Let* $G = (V, E, \lambda)$ *be an* n*-vertex weighted biconnected planar graph,* $\mathcal{E}$ *be a plane embedding of* G*, and* Γ *be a straight-line realization of* G*. There exists an* $O(n)$*-time algorithm that tests whether* Γ *is a planar straight-line realization respecting* $\mathcal{E}$.

Proof. First, we check whether Γ respects $\mathcal{E}$. This can be done in $O(n)$ time by checking: (i) whether the clockwise order of the edges incident to each vertex in Γ is the same as in $\mathcal{E}$; and (ii) whether the cycle bounding the outer face of Γ (if Γ is planar) is the same as in $\mathcal{E}$. More in detail, this check is as follows.

- Check (i) only requires to scan, for each vertex v of G, the list representing the clockwise order of the edges incident to v in $\mathcal{E}$, and to confirm that the slopes of such edges in Γ form a decreasing sequence (where the slope of one of such edges is the one of the half-ray from v through the other end-vertex of the edge and has a value between the slope s of the first considered edge and $s - 360°$). This can be done in time proportional to the degree of v and hence in $O(n)$ time over all the vertices of G.
- In order to perform check (ii), we find in $O(n)$ time the vertices $v_{\leftarrow}$ and $v_{\rightarrow}$ with smallest and largest x-coordinates, respectively, among the vertices of the cycle $\mathcal{C}$ bounding the outer

face of Γ. Then we check whether the edge that leaves $v_{\leftarrow}$ in clockwise direction along $\mathcal{C}$ lies above the edge that enters $v_{\leftarrow}$ in clockwise direction along $\mathcal{C}$ in Γ. Together with check (i), this guarantees that all the edges incident to vertices of $\mathcal{C}$ leave the vertices in $\mathcal{C}$ towards the region inside $\mathcal{C}$ in Γ, if Γ is planar. It might still be the case, however, that $\mathcal{C}$ self-intersects or that other edges of G cross $\mathcal{C}$, hence at this point we can guarantee that $\mathcal{C}$ bounds the outer face of Γ only by assuming its planarity.

In order to check whether Γ is planar, we augment Γ to a straight-line drawing Γ' of a maximal planar graph so that, if Γ is planar, then Γ' is planar, as well. This is done as follows. If Γ is a planar straight-line realization respecting $\mathcal{E}$, every internal face f of $\mathcal{E}$ is bounded in Γ by a polygon $\mathcal{P}_f$ whose interior is empty. We triangulate the interior of $\mathcal{P}_f$ by the addition of dummy edges. This can be done in linear time over all the internal faces of $\mathcal{E}$ by means of the algorithm by Chazelle [22]. We also triangulate the outer face of Γ. This is done similarly as described by Cabello et al. [18]. Namely, we enclose Γ into a large equilateral triangle T with a horizontal side ℓ. This partitions the outer face of Γ into two regions, namely a bounded region $\mathcal{R}$ and an unbounded region $\mathcal{R}'$. We then add edges from $v_{\leftarrow}$ and $v_{\rightarrow}$ to the left and right end-vertices of ℓ, respectively. This subdivides $\mathcal{R}$ into two regions delimited by simple polygons. We triangulate the interior of such polygons, again using Chazelle's algorithm [22]. Denote by Γ' the resulting straight-line drawing.

If Γ is a planar straight-line realization respecting $\mathcal{E}$, then (as described in [18]) the algorithm by Chazelle terminates correctly when applied to every polygon bounding an internal face of Γ'. Hence, if at least one of the executions of Chazelle's algorithm terminates in error, we conclude that Γ is not a planar straight-line realization respecting $\mathcal{E}$. It might also be the case that every execution of Chazelle's algorithm terminates correctly and yet Γ contained crossings. Hence, we also need to check whether Γ' is planar. This can be done in $O(n)$ time, as if Γ' is planar, it is a triangulation (i.e., a planar straight-line drawing such that every face is delimited by a triangle) and the planarity of convex subdivisions can be tested in $O(n)$ time [26]. □

Theorem 5. *Let $G = (V, E, \lambda)$ be an n-vertex weighted biconnected planar graph, let $\mathcal{R}$ be a rotation system for G, and let Γ be a straight-line realization of G. There exists an $O(n)$-time algorithm that tests whether Γ is a planar straight-line realization whose rotation system is $\mathcal{R}$.*

Proof. We are going to recover the cycle $\mathcal{C}$ that bounds the outer face of Γ (if Γ is planar) by exploiting the knowledge of $\mathcal{R}$ and of Γ itself. Then it suffices to invoke Theorem 4 in order to test whether Γ is a planar straight-line realization respecting $\mathcal{E}$, where $\mathcal{E}$ is the plane embedding that has $\mathcal{R}$ as rotation system and $\mathcal{C}$ as cycle bounding the outer face.

In order to recover the cycle $\mathcal{C}$ that bounds the outer face of Γ, we first find in $O(n)$ time the leftmost vertex v_0 in Γ and we then find in $O(n)$ time the edge (v_0, v_1) incident to v_0 with largest slope; note that, if Γ is planar, then v_0 and v_1 are encountered consecutively when traversing $\mathcal{C}$ in clockwise direction. Now suppose that, for some $i \geq 1$, a path $(v_0, v_1, \ldots, v_i)$ has been found such that, if Γ is planar and has $\mathcal{R}$ as rotation system, then the vertices $v_0, v_1, \ldots, v_i$ are encountered consecutively when traversing $\mathcal{C}$ in clockwise direction. Then, if Γ is planar and has $\mathcal{R}$ as rotation system, the vertex v_{i+1} that is encountered after v_i when traversing $\mathcal{C}$ in clockwise direction is the vertex that follows (v_i, v_{i-1}) in the clockwise order of the edges incident to v_i in $\mathcal{R}$; such a vertex can be found in $O(1)$ time. When this process encounters a vertex v_k for the second time, if such a vertex is not v_0, then we conclude that Γ is not a planar straight-line realization whose rotation system is $\mathcal{R}$, otherwise we have found the desired cycle $\mathcal{C}$. □

We remark that theorems as the previous two cannot be stated for general planar graphs. Indeed, an easy reduction from Element Uniqueness [11] shows that even testing whether a straight-line drawing of an n-vertex graph with no edges is planar requires $\Omega(n \log n)$ time.

In order to prove the main results of this section, we need the following lemma, which might be of independent interest and which is applicable to planar graphs that are not necessarily 2-trees.

Lemma 2. *Let G be an n-vertex planar graph and let $\mathcal{R}$ be a rotation system for G. A data structure can be set up in $O(n)$ time that allows one to answer in $O(1)$ time the following type of queries: Given three*

edges e_i, e_j, and e_k incident to a vertex v of G, determine whether they appear in the order e_i, e_j, and e_k or in the order e_i, e_k, and e_j in the clockwise order $\mathcal{R}(v)$ of the edges incident to v in $\mathcal{R}$.

Proof. The proof's main idea is the same as the one of Aichholzer et al. [2] for the $O(n^2)$-time construction of a data structure that allows to answer in $O(1)$ time queries about the orientation of triples of points in the plane.

Consider any vertex v of G. Let $\mathcal{R}(v) = [e_1, \ldots, e_\ell]$; note that $\mathcal{R}(v)$ is a circular list, which is here linearized by picking any edge as e_1. Label each edge e_i with the value (v, i). This labeling labels each edge of G twice, hence it is performed in $O(n)$ time. The data structure claimed in the statement simply consists of G equipped with these edge labels.

Now suppose that a query as in the statement has to be answered for three edges e_i, e_j, and e_k incident to a vertex v of G. We recover in $O(1)$ time the labels (v, i), (v, j), and (v, k) respectively associated to e_i, e_j, and e_k. Assume that i is smaller than j and k; then e_i, e_j, and e_k appear in the order e_i, e_j, and e_k in $\mathcal{R}(v)$ if and only if $j < k$, which can be checked in $O(1)$ time. Analogously, if $j < i$ and $j < k$, we have that e_i, e_j, and e_k appear in the order e_i, e_j, and e_k in $\mathcal{R}(v)$ if and only if $k < i$, while if $k < i$ and $k < j$, we have that e_i, e_j, and e_k appear in the order e_i, e_j, and e_k in $\mathcal{R}(v)$ if and only if $i < j$. □

We now present the main results of this section.

Theorem 6. *Let G be an n-vertex weighted 2-tree and $\mathcal{R}$ be a rotation system for G. There exists an $O(n)$-time algorithm that tests whether G admits a planar straight-line realization whose rotation system is $\mathcal{R}$ and, in the positive case, constructs such a realization.*

Proof. The algorithm is as follows. First, among all the 3-cycles of G, we pick a 3-cycle c with maximum area, which can clearly be done in $O(n)$ time. Second, we compute in $O(n)$ time the decomposition tree T of G rooted at c. Third, by means of Lemma 2, we set up in $O(n)$ time a data structure that allows us to decide in $O(1)$ time whether any three edges e_i, e_j, and e_k incident to a vertex v of G appear in the order e_i, e_j, and e_k or in the order e_i, e_k, and e_j in the clockwise order of the edges incident to v in $\mathcal{R}$.

Let γ be any straight-line realization of c and let γ' be the reflection of γ. Note that γ is unique, up to rigid transformations. Hence, if there exists a planar straight-line realization Δ of G whose rotation system is $\mathcal{R}$ and whose restriction to c is some triangle δ, then there also exists a planar straight-line realization Γ of G whose rotation system is $\mathcal{R}$ and whose restriction to c is either γ or γ'. Indeed, Γ can be obtained from Δ by applying the rotation and translation that turn δ into either γ or γ'; then the rotation system for Γ is the same as the one for Δ, hence it is $\mathcal{R}$.

We show how to test in $O(n)$ time whether there exists a planar straight-line realization of G whose rotation system is $\mathcal{R}$ and whose restriction to c is γ (in the positive case, such a realization is constructed within the same time bound). An analogous test can be performed with the constraint that c is represented by γ' rather than γ. Then the algorithm concludes that a planar straight-line realization of G whose rotation system is $\mathcal{R}$ exists if and only if (at least) one of the two tests is successful.

We are going to use the following key property.

Property 1. *Let (u, v, w) be a cycle of G such that u is not a vertex of c. Let P be a shortest path from u to (any vertex of) c and suppose that the edge of P incident to u, say (u, z), is such that z is different from v and w. Then, in any planar straight-line realization of G, the edge (u, z) lies outside (u, v, w).*

Proof. Consider any planar straight-line realization Γ of G. Since z is different from v and w and since P is a shortest path, we have that P contains neither v nor w. Hence, if z lies inside (u, v, w), then so does c, by the planarity of Γ. However, this is not possible, since c is a 3-cycle whose area is maximum, by definition. □

Our algorithm visits T in pre-order starting at c; let $c_1, \ldots, c_{n-2}$ be the order in which the nodes of T are visited. For $i = 1, \ldots, n-2$, let G_i be the subgraph of G composed of the 3-cycles $c_1, \ldots, c_i$; note that $G_1 = c = c_1$ and $G_{n-2} = G$. Also, let $\mathcal{R}_i$ be the restriction of $\mathcal{R}$ to G_i.

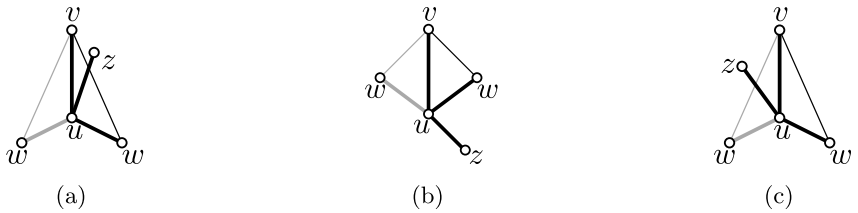

Fig. 4. Illustrations of Cases 1, 2, and 3 for the insertion of w and its incident edges in Γ_i. Both possible placements of w are shown (a chosen placement is shown in black).

If Γ is a planar straight-line realization of G whose rotation system is $\mathcal{R}$ and whose restriction to c is γ, then, for any $i = 1, \ldots, n-2$, the restriction of Γ to G_i is a planar straight-line realization Γ_i of G_i whose rotation system is $\mathcal{R}_i$ and whose restriction to c is γ. We prove that, if such a realization Γ_i of G_i exists, then it is unique and can be computed in $O(1)$ time from Γ_{i-1} (from γ, if $i = 1$). During the visit of T we maintain, for each vertex u of G_i that does not belong to c, the first edge (u, z) of a shortest path P_u from u to (any vertex of) c in G_i. Further, we maintain, for each vertex u of G_i, a value $\ell(u)$; this is equal to the number of edges of P_u, if u does not belong to c, and to 0, otherwise.

When visiting c_1, we set $\Gamma_1 = \gamma$ in $O(1)$ time. Obviously, Γ_1 is the unique planar straight-line realization of G_1 in which the straight-line realization of c_1 is γ (the rotation system $\mathcal{R}_1$ does not impose any constraint). We initialize $\ell(u) = 0$ for each vertex u of c_1.

Assume that, for some $i \in \{1, \ldots, n-3\}$, a straight-line realization Γ_i of G_i has been constructed such that, if a planar straight-line realization whose rotation system is $\mathcal{R}_i$ and whose restriction to c is γ exists, then Γ_i is the unique such a realization. Let $c_{i+1} = (u, v, w)$, where $e_{uv} := (u, v)$ is the edge that c_{i+1} shares with the 3-cycle represented by the parent of c_{i+1} in T, while w is the unique vertex of G_{i+1} not in G_i. Let e_{uw} and e_{vw} denote the edges (u, w) and (v, w), respectively. Assume, w.l.o.g., that $\ell(u) \leq \ell(v)$. Then we compute in $O(1)$ time the value $\ell(w)$ as $\ell(w) = 1 + \ell(u)$ (the $\ell(\cdot)$ value of every other vertex of G_{i+1} coincides with the one in G_i). Also, again in $O(1)$ time, we set e_{uw} as the first edge of a shortest path from w to c (for every other vertex of G_{i+1}, the first edge of a shortest path to c coincides with the one in G_i).

Note that there are two placements for w that result in the edges e_{uw} and e_{vw} having the prescribed edge lengths, thus turning Γ_i into a straight-line realization of G_{i+1}. We show that Property 1 can be used in order to define a unique placement for w in Γ_i.

If $\ell(u) > 0$, let $e_{uz} := (u, z)$ be the first edge of the shortest path from u to c in G_i (recall that we maintain this information). If $\ell(u) = 0$, then u is a vertex of c; then we define $e_{uz} := (u, z)$ as any edge of c different from e_{uv} (note that e_{uv} might or might not be an edge of c). In both cases, we use Lemma 2 to determine in $O(1)$ time whether the clockwise order of the edges e_{uv}, e_{uw}, and e_{uz} in $\mathcal{R}$ is e_{uv}, e_{uw}, and e_{uz} or e_{uv}, e_{uz}, and e_{uw}. Suppose, without loss of generality, it is the former. We now proceed according to the following case distinction.

Case 1: If neither of the two possible placements of w guarantees that the clockwise order of the edges e_{uv}, e_{uw}, and e_{uz} in Γ_{i+1} is e_{uv}, e_{uw}, and e_{uz}, as in Fig. 4(a), then we conclude that there exists no planar straight-line realization of G_{i+1} whose rotation system is $\mathcal{R}_{i+1}$ and whose restriction to c is γ.

Case 2: If exactly one of the two possible placements of w guarantees that the clockwise order of the edges e_{uv}, e_{uw}, and e_{uz} in Γ_{i+1} is e_{uv}, e_{uw}, and e_{uz}, as in Fig. 4(b), then we choose that placement for w. Indeed, choosing the other placement for w would imply that, even if the resulting straight-line realization of G_{i+1} was planar, it would not have $\mathcal{R}_{i+1}$ as its rotation system.

Case 3: If both the possible placements of w guarantee that the clockwise order of the edges e_{uv}, e_{uw}, and e_{uz} in Γ_{i+1} is e_{uv}, e_{uw}, and e_{uz}, as in Fig. 4(c), then we observe that one of the two placements of w is such that (at least part of) the edge e_{uz} is internal to the triangle

(u, v, w), while the other placement is such that the edge e_{uz} is external to the triangle (u, v, w). We choose the latter placement for w.
If $\ell(u) > 0$, this choice is motivated by Property 1. Indeed, u is not a vertex of c, given that $\ell(u) > 0$; further, z is different from w (as z belongs to G_i, while w does not) and from v (by the assumption $\ell(u) \leq \ell(v)$ and $\ell(z) = \ell(u) - 1$). Hence, Property 1 applies and in any planar straight-line realization of G_{i+1}, the edge e_{uz} is external to (u, v, w).
If $\ell(u) = 0$, then e_{uz} is an edge of c, hence if e_{uz} were internal to (u, v, w), then c would be internal to (u, v, w), as well (the two cycles share the vertex u and, possibly, the vertex v and the edge (u, v)). However, this is not possible in any planar straight-line realization of G_{i+1}, as the area of c is greater than or equal to the area of (u, v, w), by the definition of c.

The clockwise order of the edges e_{uv}, e_{uw}, and e_{uz} according to a placement for w can be computed in $O(1)$ time, as well as whether the edge e_{uz} is external to the triangle (u, v, w) or not. Hence, Γ_{i+1} can be constructed in $O(1)$ time from Γ_i.

When $i = n-2$, we have that, if there exists a planar straight-line realization of $G = G_{n-2}$ whose rotation system is $\mathcal{R} = \mathcal{R}_{n-2}$ and whose restriction to c is γ, then the constructed representation $\Gamma := \Gamma_{n-2}$ is the unique such a realization. Hence, it only remains to test whether Γ is a planar straight-line realization of G whose rotation system is $\mathcal{R}$. This can be done in $O(n)$ time by means of Theorem 5. □

We can similarly deal with the case in which the cycle bounding the outer face of the planar straight-line realization is prescribed, as in the following theorem.

Theorem 7. *Let G be an n-vertex weighted 2-tree and $\mathcal{E}$ be a plane embedding of G. There exists an $O(n)$-time algorithm that tests whether G admits a planar straight-line realization that respects $\mathcal{E}$ and, in the positive case, constructs such a realization.*

Proof. Let $\mathcal{R}$ be the rotation system associated with $\mathcal{E}$. By exploiting $\mathcal{R}$ and ignoring the knowledge of the cycle delimiting the outer face of $\mathcal{E}$, at most two straight-line realizations of G can be constructed in $O(n)$ time, exactly as in the proof of Theorem 6, such that G admits a planar straight-line realization whose rotation system is $\mathcal{R}$ if and only if one of these two realizations is planar and has $\mathcal{R}$ as its rotation system. Differently from the proof of Theorem 6, we use Theorem 4 rather than Theorem 5 to test in $O(n)$ time whether any of the two realizations is planar and respects $\mathcal{E}$. □

4. NP-hardness for 2-trees with four edge lengths

In this section, we present a polynomial-time reduction from the Planar Monotone 3-SAT problem to the FEPR problem with *four* edge lengths. As the former is known to be NP-complete [12], the reduction shows NP-hardness of the latter. We thus have the following.

Theorem 8. *The* Fixed Edge-Length Planar Realization *problem is NP-hard for weighted 2-trees, even for instances whose number of distinct edge lengths is 4.*

Let ϕ be a Boolean formula in conjunctive normal form with at most three literals in each clause. We denote by G_ϕ the *incidence graph* of ϕ, that is, the graph that has a vertex for each clause of ϕ, a vertex for each variable of ϕ, and an edge (c_i, v_j) for each clause c_i that contains the literal v_j or the literal $\overline{v}_j$. We remark that vertices representing clauses have degree 3 in G_ϕ, whereas vertices representing variables might have degree larger than 3. The formula ϕ is an instance of Planar Monotone 3-SAT if G_ϕ is planar and each clause of ϕ is either *positive* or *negative*. A positive clause contains only *positive literals* (i.e., literals v_j for some variable v_j), while a negative clause only contains *negated literals* (i.e., literals $\overline{v}_j$ for some variable v_j). The Planar Monotone 3-SAT problem asks whether there exists a truth assignment for the variables of ϕ such that each clause has at least a `true` literal.

Let ϕ be an instance of Planar Monotone 3-SAT. Hereafter we assume that each clause of ϕ contains *exactly* three literals. This is not a loss of generality, since a clause with less than three

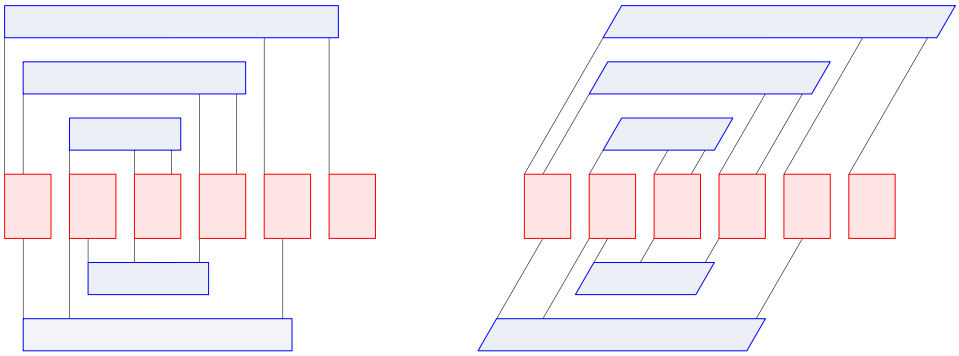

Fig. 5. Monotone rectilinear representations of G_ϕ. Boxes representing variables are red, whereas boxes and trapezoids representing clauses are blue. (a) The original monotone rectilinear representation Γ_ϕ. (b) The modified monotone rectilinear representation Γ^*_ϕ that satisfies properties D1–D7. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

literals can be modified by duplicating one of the literals in the clause, which does not alter the satisfiability of ϕ. Note this modification might turn G_ϕ into a planar multi-graph. Nevertheless, since this is not relevant for our reduction, it will be ignored hereforth.

A *monotone rectilinear representation* of G_ϕ is a drawing that satisfies the following properties (refer to Fig. 5(a)):

P1: Variables and clauses are represented by axis-aligned boxes with the same height.
P2: The bottom sides of all boxes representing variables lie on the same horizontal line.
P3: The boxes representing positive (resp. negative) clauses lie above (resp. below) the boxes representing variables.
P4: The edges connecting variables and clauses are vertical segments.
P5: The drawing is crossing-free.

The Planar Monotone 3-SAT problem has been shown to be NP-complete, even when the incidence graph is provided along with a monotone rectilinear representation [12]. Given an instance ϕ of the Planar Monotone 3-SAT problem and a rectilinear representation Γ_ϕ of ϕ, we construct a weighted 2-tree H_ϕ that admits a planar straight-line realization *if and only if* ϕ is satisfiable. The general strategy of the reduction is as follows. In Section 4.1 we describe how to modify Γ_ϕ into an auxiliary representation Γ^*_ϕ that satisfies a set of convenient properties (see Fig. 5(b)). We then exploit the geometric information of Γ^*_ϕ to construct H_ϕ, so that the edges of H_ϕ are assigned four distinct lengths. Namely, in Section 4.3, we describe gadgets for the variables, for the clauses, and for the edges of G_ϕ. Finally, in Section 4.4, we show how to combine these gadgets to form H_ϕ.

4.1. The auxiliary monotone rectilinear representation

We start by describing how to transform Γ_ϕ into a new representation Γ^*_ϕ that satisfies the following properties (refer to Fig. 5(b)):

D1: The corners of the polygons and the end-points of the segments forming Γ^*_ϕ lie on the points of the grid formed by equidistant lines with slope 0°, 60°, and 120°, in which each grid cell is an equilateral triangle with sides of unit length.
D2: The height and width of the bounding box of Γ^*_ϕ are polynomially bounded in the size of ϕ.
D3: The variables are represented by axis-aligned boxes. Let δ_ϕ denote the maximum degree of G_ϕ. The boxes have width $2\delta_\phi - 1$, height $2\sqrt{3}(\delta_\phi - 1)$, and have their bottom sides on a common horizontal grid line.

D4: The clauses are represented by trapezoids. Each trapezoid has height $2\sqrt{3}$, lateral sides with slope 60°, and horizontal sides whose length is larger than 8.

D5: The edges connecting variables and clauses are line segments with slope 60°.

D6: Consider the box $\mathcal{B}$ representing a variable. The edges incident to $\mathcal{B}$ have horizontal distance from the left side of $\mathcal{B}$ that is a multiple of 2.

D7: Consider the trapezoid $\mathcal{T}$ representing a positive (resp. negative) clause c. Let p_1, p_2, and p_3 be the intersection points between the segments representing the edges of G_ϕ incident to c and the bottom (resp. top) horizontal side of $\mathcal{T}$. The point p_1 lies on the bottom-left (resp. top-left) corner of $\mathcal{T}$, the horizontal distance between p_1 and p_2 is at least three, the horizontal distance between p_2 and p_3 is at least four, and the horizontal distance between p_3 and the bottom-right (resp. top-right) corner of $\mathcal{T}$ is equal to one.

The transformation is described in the following lemma.

Lemma 3. *The drawing Γ_ϕ^* can be constructed in polynomial time starting from Γ_ϕ.*

Proof. We first transform Γ_ϕ into an intermediate rectilinear representation Γ_ϕ' that satisfies properties D1–D3, property D6, and two properties that we denote by **D4'** and **D7'**, obtained from properties D4 and D7, respectively, by substituting trapezoids with boxes. Since every segment of a geometric object forming Γ_ϕ is either horizontal or vertical, we can transform Γ_ϕ into Γ_ϕ' by suitable scaling the boxes representing variables and clauses, and by a suitable deformation of the plane.

We transform Γ_ϕ' into Γ_ϕ^* as follows. Observe that, since Γ_ϕ' satisfies property D3, the top (resp. bottom) sides of the boxes representing variables lie on a horizontal line ℓ_t (resp. ℓ_b). The transformation consists of two horizontal shear mappings that transform vertical segments into segments with slope 60° and are applied, respectively, to the points of the plane above ℓ_t and below ℓ_b. More precisely, these shear mappings are as follows. Translate the Cartesian reference system so that the x-axis coincides with ℓ_t (and the origin is anywhere). The first shear mapping then maps every point (x, y) with $y \geq 0$ to the point $(x + \frac{\sqrt{3}}{2}y, y)$. Again translate the Cartesian reference system so that the x-axis coincides with ℓ_b (and the origin is anywhere). The second shear mapping then maps every point (x, y) with $y \leq 0$ to the point $(x - \frac{\sqrt{3}}{2}y, y)$.

We show next that the resulting representation Γ_ϕ^* satisfies properties D1–D7:

D1**:** This property is satisfied since Γ_ϕ' satisfies property D1. Note that the distance between two consecutive grid lines (with slope either 0°, or 60°, or 120°) is $\frac{\sqrt{3}}{2}$.

D2**:** Let $\mathcal{B}'$ and $\mathcal{B}^*$ denote, respectively, the bounding boxes of Γ_ϕ' and Γ_ϕ^*. Property D2 is satisfied since Γ_ϕ' satisfies property D2, the height of $\mathcal{B}^*$ is equal to the height of $\mathcal{B}'$, and the width of $\mathcal{B}^*$ is at most the width of $\mathcal{B}'$ plus $\sqrt{3}$ times the height of $\mathcal{B}'$.

D3 **and** D6**:** These properties are satisfied since Γ_ϕ' satisfies properties D3 and D6, and the shear mappings do not move the points contained in the horizontal strip bounded by ℓ_t and ℓ_b.

D4**,** D5**, and** D7**:** These properties are satisfied since Γ_ϕ' satisfies properties D4' and D7', since the shear mappings transform vertical segments outside the horizontal strip delimited by ℓ_t and ℓ_b into segments with slope 60°, and since the shear mappings do not alter the distance between any two points on the same horizontal line.

We complete the proof by observing that all the transformations described above can be computed in polynomial time. □

4.2. Overview of the reduction

For the sake of clarity, before describing in detail the reduction from Γ_ϕ^* to H_ϕ, we provide next a high-level description of the gadgets we employ to obtain H_ϕ. In Section 4.3 we present complete constructions, as well as lemmas to guarantee that the gadgets behave as required for the correctness of the reduction.

We define three main types of gadgets. A variable $v_i \in \phi$ is modeled by means of a gadget we call *variable gadget*, a clause $c_j \in \phi$ by means of a gadget we call a *clause gadget*, and an edge $(v_i, c_j) \in G_\phi$ by means of a gadget we call a *k-transmission gadget*. We also define two auxiliary gadgets we call *ladder gadgets* and *flag gadgets*. Ladder gadgets are used to provide structural rigidity to the reduction, and flag gadgets are used as an auxiliary tool in the construction of clause gadgets. To construct the gadgets we use the lengths $w_1 := 1$, $w_2 := 0.9$, $w_3 := 0.2$, and $w_4 := 1.61$. In the figures, we represent edges of length w_1, w_2, w_3, and w_4 by means of black, red, blue, and green segments, respectively. The lengths w_1 and w_2 play a special role in the reduction, which is described below; conversely, the lengths w_3 and w_4 are only used in clause gadgets. We use w_1 and w_2 to define two main types of triangles: Equilateral triangles with sides of length w_1, which we call *frame triangles*, and flat isosceles triangles with base of length w_1 and two shorter sides of length w_2, which we call *transmission triangles*. We refer to the subgraph of a gadget formed by its frame triangles as the *frame* of the gadget. With the exception of clause gadgets, the frame of every gadget is a maximal outerplanar graph. Then the frame of every gadget other than the clause gadget has a unique planar straight-line realization (up to rigid transformations), given that no frame triangle can lie inside a distinct frame triangle, as these triangles are equilateral and have the same side length. The frame of a clause gadget consists of three maximal outerplanar graphs, each of which has a unique planar straight-line realization (up to rigid transformations). The variable, clause, and *k*-transmission gadgets are constructed in such a way that every frame triangle shares two of its sides each with a different transmission triangle. Since $\frac{w_1}{w_2} = \frac{1}{0.9} < \sqrt{3}$, by Lemma 1 we have that, in any planar straight-line realization of H_ϕ, two transmission triangles that share a side with a frame triangle cannot both be drawn inside such a frame triangle. This property is exploited in order to propagate along the frame of the gadgets the truth values of each literal; this value is initially determined by the embedding of a transmission triangle inside one or the other of its two incident frame triangles of a variable gadget. A final common property of all our gadgets is a set of special edges we call *attachment edges*, which are edges of frame triangles that are also edges of transmission triangles. In the figures, attachment edges are represented as thick black segments. We use attachment edges to combine gadgets together so that the encoded truth values can be propagated along the resulting 2-tree. We exploit properties D1–D7 of Γ^*_ϕ to guarantee that, after combining all the gadgets, H_ϕ has a planar straight-line realization if and only if ϕ is satisfiable.

An example of a planar straight-line realization of H_ϕ is shown in Fig. 6. This example illustrates a high-level sketch of the reduction from Γ^*_ϕ to H_ϕ. The grid formed by lines with slope 0°, 60°, and 120° is shown in light gray. For the sake of clarity, we only show the frames of the gadgets forming H_ϕ. To illustrate the correspondence with the boxes of Fig. 5(b) that represent the variables and clauses of ϕ, we also show the bounding boxes of the variable gadgets (in dashed red) and the trapezoids enclosing the clause gadgets (in dashed blue). We next provide a high-level description of each gadget.

The variable gadget. Remember that δ_ϕ denotes the maximum degree of G_ϕ. The variable gadget provides $2\delta_\phi$ attachment edges. Further, its frame is a maximal outerplanar graph, hence it has a unique planar straight-line realization, up to rigid transformations. Let Γ be a planar straight-line realization of the variable gadget, $\Gamma_f \subset \Gamma$ be the realization of the frame of the gadget in Γ, and B_f be the bounding box of Γ_f. In Fig. 6, the realization Γ_f is shaded gray and B_f is shown with red dashed lines. Up to a rotation of Γ, we have that δ_ϕ attachment edges lie along the top side of B_f, say s_t, while the remaining δ_ϕ attachment edges lie along the bottom side of B_f, say s_b. The crucial property of the variable gadget is that, in Γ, either *all* the transmission triangles incident to attachment edges along s_t are drawn outside Γ_f, or *all* the transmission triangles incident to attachment edges along s_b are drawn outside Γ_f. We exploit this property to encode the truth assignment of the variable modeled by the gadget. In particular, the purpose of the attachment edges on s_t is to propagate the truth value to incident positive clauses, while the purpose of the attachment edges on s_b is to propagate the negated truth value to incident negative clauses.

The ladder gadget. By Property D3, the bottom sides of the boxes representing variables lie on the same horizontal grid line in Γ^*_ϕ. Consider the order of appearance of the boxes while traversing such line from left to right. The ladder gadget consists of a sequence of frame triangles that form

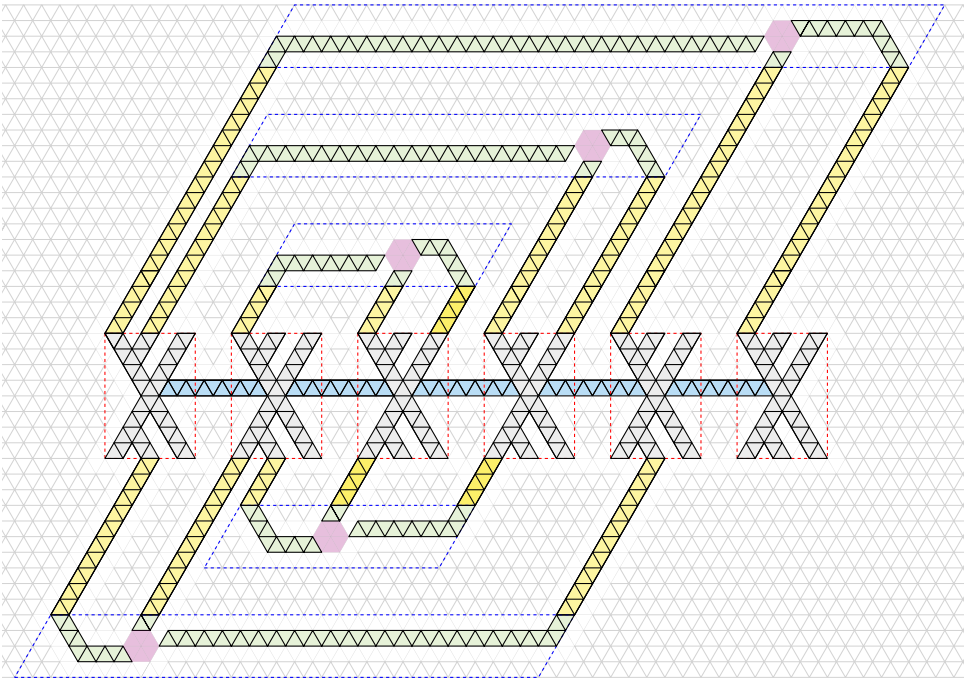

Fig. 6. Overview of the reduction showing only the frame triangles of the involved gadgets. The frame of each variable gadget is shaded light gray and is enclosed by a dashed red rectangle. The frame of each ladder gadget is shaded blue. The frame of each transmission gadget that is not part of a clause gadget is shaded yellow. The frame of each connected component of the clause gadget is shaded light green. Each clause gadget is enclosed by a dashed blue trapezoid. The pink region inside a clause gadget is the region where the planarity of the realization can be guaranteed if and only if at least one of the literals of the corresponding clause is satisfied. The subgraph of H_ϕ that is embedded inside each pink region is not connected, indeed it is composed of three connected components, representing the three literals in the clause. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

a maximal outerpath and that connect the frames of two consecutive variable gadgets; see the triangles shaded blue in Fig. 6. The purpose of the ladder gadgets is to make the entire graph H_ϕ biconnected and to make the union of the frames of all the gadgets a maximal outerplanar graph, which has a unique planar straight-line realization, up to rigid transformations. This realization can then be navigated by the truth values of the variables.

The k-transmission gadget. Let v_i and c_j be a variable and a clause of ϕ, respectively, such that (v_i, c_j) is an edge of G_ϕ. Let $\mathcal{V}_i$ denote the variable gadget modeling v_i and $\mathcal{C}_j$ denote the clause gadget modeling c_j. The purpose of the transmission gadget is to "transmit" to $\mathcal{C}_j$ the truth value of v_i corresponding to the realization of $\mathcal{V}_i$. The frame of the k-transmission gadget is a sequence of k frame triangles that form a maximal outerpath; see the triangles shaded yellow in Fig. 6. The parameter k coincides with twice the length of the segment representing the edge (v_i, c_j) in Γ^*_ϕ. The k-transmission gadget provides two attachment edges, each incident to one of the faces of the frame that correspond to the end-vertices of its dual path. To connect $\mathcal{V}_i$ to $\mathcal{C}_j$, one of these attachment edges is identified with an attachment edge of $\mathcal{V}_i$ and the second one is identified with an attachment edge of $\mathcal{C}_j$.

The clause gadget. Unlike the aforementioned gadgets, the clause gadget consists of three distinct connected components. Two of these components are formed by frame and transmission triangles. The third component is not only formed by frame and transmission triangles, but also by a special

subgraph we call a *flag gadget*. The name of this gadget is due to the resemblance of (some of) its realizations with a flag in the wind. The frame of each component forms a maximal outerpath; see the triangles shaded light green in Fig. 6. The clause gadget contains six attachment edges. Three of them, that we call *input attachment edges*, are used to combine the clause gadget with the three k-transmission gadgets modeling the edges of G_ϕ incident to the clause. The remaining three, that we call *output attachment edges*, are used to model the logic of the clause; in particular, one of them connects the frame of a component to the flag gadget.

Consider a positive clause c_j that contains the literal v_i. The clause c_j is modeled in H_ϕ by a clause gadget $\mathcal{C}_j$ and is represented in Γ_ϕ^* by a trapezoid t_j. The edge (v_i, c_j) of G_ϕ is modeled in H_ϕ by a k-transmission gadget $\mathcal{T}_{i,j}$ and is represented in Γ_ϕ^* by a line segment $s_{i,j}$. The gadgets $\mathcal{C}_j$ and $\mathcal{T}_{i,j}$ are combined together by means of an attachment edge e of $\mathcal{T}_{i,j}$ that is identified with an input attachment edge of $\mathcal{C}_j$. The combination of $\mathcal{C}_j$ with $\mathcal{T}_{i,j}$ exploits the following geometric property. Consider the unique planar straight-line realization of the subgraph of H_ϕ that is the union of all the ladder gadgets and of the frames of all the variable and transmission gadgets. By Property D5 of Γ_ϕ^* and the fact that k coincides with twice the length of $s_{i,j}$, we have that, in such a realization, the attachment edge e is represented by a line segment lying along the bottom side of t_j. If c_j is instead negative, then e lies along the top side of t_j.

By the geometric property described above, the input attachment edges of the components of $\mathcal{C}_j$ are represented by line segments that lie either all on the bottom or all on the top side of t_j. The frames of the components of $\mathcal{C}_j$ can thus be defined so that, in any planar straight-line realization of $\mathcal{C}_j$, we have that (i) all the frames are entirely contained in t_j and (ii) the output attachment edges are represented by line segments that are "close" to each other. In Fig. 6, we emphasize this closeness by depicting a pink-shaded hexagon whose vertices coincide with some of the endpoints of these line segments.

Consider the transmission triangles incident to the output attachment edges of the components of $\mathcal{C}_j$. The clause gadget admits a planar straight-line realization *if and only if* at least one of these transmission triangles is drawn inside its incident frame triangle. Let C be a component of $\mathcal{C}_j$. Assume again that the input attachment edge of C is shared with a k-transmission gadget $\mathcal{T}_{i,j}$ that connects $\mathcal{C}_j$ to the variable gadget representing a variable v_i. Our gadgets guarantee that, in any planar straight-line realization of H_ϕ, the transmission triangle incident to the output attachment edge of C can be drawn inside its adjacent frame triangle only if the gadget modeling v_i represents the value `True` (resp. `False`) and c_j is positive (resp. negative). Thus, if all three variables appearing in c_j represent the value `False` (resp. `True`) and c_j is positive (resp. negative), then H_ϕ admits no planar straight-line realization. Conversely, if at least one of the variables appearing in c_j represents the value `True` (resp. `False`) and c_i is positive (resp. negative), then the transmission triangle of the corresponding component of $\mathcal{C}_j$ can be drawn inside its incident frame triangle so that $\mathcal{C}_j$ admits a planar straight-line realization.

4.3. Description of the gadgets

We now present complete constructions for the following gadgets: The *transmission gadget*, the *split gadget*, the *variable gadget*, the *flag gadget*, and the *clause gadget*. We will later show how to precisely combine these gadgets to obtain H_ϕ.

In the following, whenever we combine two gadgets G_1 and G_2, we perform a 2-clique-sum of G_1 and G_2 as follows. First, we identify one attachment edge of G_1 with one attachment edge of G_2 to obtain a single edge e that belongs to both G_1 and G_2. Second, we remove the degree-2 vertex of any of the two transmission triangles sharing the edge e, together with the two edges incident to such vertex. This operation always results in a 2-tree, as stated in the next observation, which follows from the fact that the operation described in the statement is a 2-clique-sum and from the fact that any 2-tree is defined as a sequence of 2-clique-sums of 3-cycles.

Observation 9. *Let G_1 and G_2 be two 2-trees, and let e_1 be an edge of G_1 and e_2 be an edge of G_2. The graph G obtained from G_1 and G_2 by identifying e_1 and e_2 is a 2-tree.*

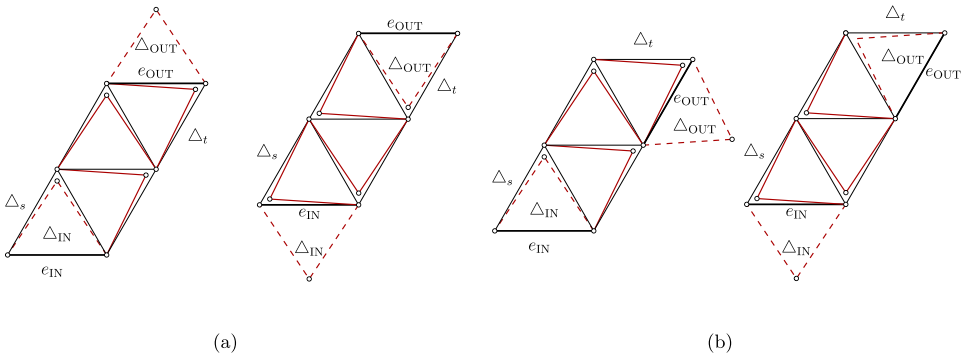

Fig. 7. Illustration of two k-transmission gadgets (in (a) and in (b)), differing from one another for the choice of the attachment edge belonging to $\triangle_t$. For each of the two k-transmission gadgets, both a planar straight-line realization in which $\triangle_{\text{IN}}$ is drawn inside $\triangle_s$ and a planar straight-line realization in which $\triangle_{\text{OUT}}$ is draw inside $\triangle_t$ are shown. In these illustrations, we have $k = 4$.

The k-transmission gadget. For any positive even integer k, a *k-transmission gadget* consists of a sequence $\triangle_s = \triangle_1, \ldots, \triangle_k = \triangle_t$ of k frame triangles, in which each triangle shares exactly one edge with its successor; refer to Fig. 7. For each pair of consecutive frame triangles, we insert a transmission triangle (solid red triangles in Fig. 7) whose base edge we identify with the common edge between the frame triangles. When the value k is not relevant, we refer to a k-transmission gadget simply as a "transmission gadget". A k-transmission gadget provides two attachment edges. The first attachment edge, which we denote e_{IN}, can be selected as any of the two edges of $\triangle_s$ that is not shared with its successor frame triangle. Similarly, the second attachment edge, which we denote e_{OUT}, can be selected as any of the two edges of $\triangle_t$ that is not shared with its predecessor frame triangle. This yields four possible k-transmission gadgets. We complete the construction of a transmission gadget by inserting a transmission triangle $\triangle_{\text{IN}}$ whose base we identify with e_{IN}, and a transmission triangle $\triangle_{\text{OUT}}$ whose base we identify with e_{OUT}; see the dashed triangles in Fig. 7.

Note that a transmission gadget is a 2-tree and its frame is a maximal outerpath. In any planar straight-line realization of a transmission gadget, its attachment edges are either parallel or not. In the first case, we say that the transmission gadget is *straight*, whereas in the second case we say that it is *turning*; the transmission gadgets in Figs. 7(a) and 7(b) are straight and turning, respectively. We have the following lemma.

Lemma 4. *In any planar straight-line realization of a transmission gadget, if $\triangle_{IN}$ is drawn inside $\triangle_s$, then $\triangle_{OUT}$ is drawn outside $\triangle_t$. Further, if $\triangle_{OUT}$ is drawn inside $\triangle_t$, then $\triangle_{IN}$ is drawn outside $\triangle_s$.*

Proof. Consider the sequence $\triangle_1, \ldots, \triangle_k$ of frame triangles of a transmission gadget, where $\triangle_s = \triangle_1$ and $\triangle_t = \triangle_k$. For $i = 1, \ldots, k-1$, the triangles $\triangle_i$ and $\triangle_{i+1}$ share a single edge, which is the base of a transmission triangle we denote with $\triangle_i^T$. We only prove the first part of the statement, since the second part is symmetric. Consider any planar straight-line realization of a transmission gadget in which $\triangle_{\text{IN}}$ is drawn inside $\triangle_s$.

Since $\frac{w_1}{w_2} = \frac{1}{0.9} < \sqrt{3}$, by Lemma 1 we have that $\triangle_{\text{IN}}$ and $\triangle_1^T$ are not both drawn inside $\triangle_1 = \triangle_s$; therefore $\triangle_1^T$ is drawn inside $\triangle_2$. Similarly, since $\triangle_i^T$ is drawn inside $\triangle_{i+1}$, then $\triangle_{i+1}^T$ is drawn inside $\triangle_{i+2}$ for $i = 1, \ldots, k-2$. Finally, since $\triangle_{k-1}^T$ is drawn inside $\triangle_k = \triangle_t$, we have that $\triangle_{\text{OUT}}$ is drawn outside $\triangle_t$. □

The split gadget. Let T^1 and T^2 be two straight 4-transmission gadgets. The split gadget can be obtained from T^1 and T^2 as follows. For $i = 1, 2$, let us denote the triangles $\triangle_{\text{IN}}$, $\triangle_{\text{OUT}}$, $\triangle_s$, and $\triangle_t$ of T^i as $\triangle_{\text{IN}}^i$, $\triangle_{\text{OUT}}^i$, $\triangle_s^i$, and $\triangle_t^i$, respectively, and the attachment edges e_{IN} and e_{OUT} of T^i as e_{IN}^i and e_{OUT}^i, respectively; see Fig. 8(a). For $i = 1, 2$, let $\triangle_i'$ be the transmission triangle of T^i that is incident

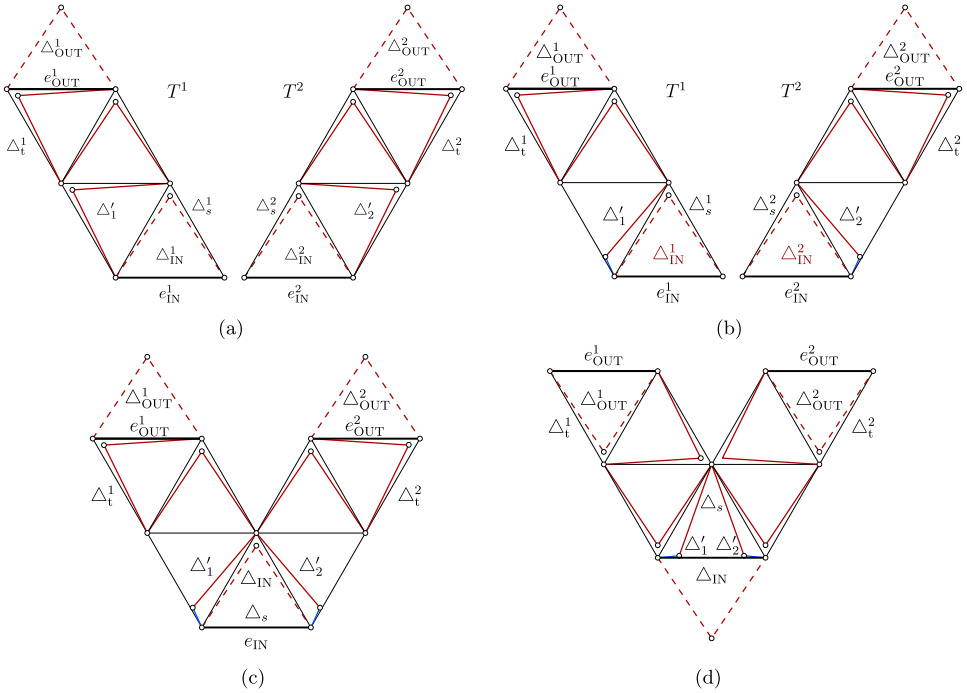

Fig. 8. Split gadget construction. (a) The 4-transmission gadgets T^1 and T^2. (b) The gadgets T^1 and T^2 after replacing $\triangle'_1$ and $\triangle'_2$ with scalene triangles. (c) The split gadget with $\triangle_{\text{IN}}$ drawn inside $\triangle_s$. (d) The split gadget with $\triangle^i_{\text{OUT}}$ drawn inside $\triangle^i_t$, for $i = 1, 2$.

to the frame triangle $\triangle^i_s$ and different from $\triangle^i_{\text{IN}}$. First, we replace $\triangle'_i$ with a scalene triangle, which we still refer to as $\triangle'_i$, whose longest side is shared with $\triangle^i_s$ and whose two shorter sides have length w_2 and w_3; refer to Fig. 8(b). After this replacement, we still denote the "modified" transmission gadgets as T^1 and T^2. Finally, we identify the frame triangles $\triangle^1_s$ and $\triangle^2_s$ to obtain a single frame triangle we denote with $\triangle_s$, and we identify the transmission triangles $\triangle^1_{\text{IN}}$ and $\triangle^2_{\text{IN}}$ to obtain a single transmission triangle we denote with $\triangle_{\text{IN}}$; refer to Figs. 8(c) and 8(d). By Observation 9 and the fact that the transmission gadgets are 2-trees, we have that the split gadget is a 2-tree and its frame is a maximal outerpath.

The split gadget provides three attachment edges. Namely, the attachment edge e_{IN} incident to $\triangle_{\text{IN}}$ (obtained by identifying e^1_{IN} and e^2_{IN}), the attachment edge e^1_{OUT} incident to $\triangle^1_{\text{OUT}}$, and the attachment edge e^2_{OUT} incident to $\triangle^2_{\text{OUT}}$. We have the following property.

Property 2. *The scalene triangles $\triangle'_1$ and $\triangle'_2$ satisfy the following properties.*

(a) The triangles $\triangle'_1$ and $\triangle'_2$ can be drawn together inside $\triangle_s$ without intersecting each other.
(b) Neither $\triangle'_1$ nor $\triangle'_2$ can be drawn together with $\triangle_{IN}$ inside $\triangle_s$ without intersecting each other.

Proof. Statement (a) follows from the fact that the angles of $\triangle'_1$ and $\triangle'_2$ incident to the vertex they share sum up to less than 60°. Indeed, by the law of the cosines, such angles are equal to $\arccos\left(\frac{w_1^2+w_2^2-w_3^2}{2w_1w_2}\right) < \arccos(0.983) < 11°$.

Statement (b) follows from the fact that, for $i = 1, 2$, the angles of $\triangle'_i$ and $\triangle_{\text{IN}}$ incident to the vertex they share sum up to more than 60°. Indeed, by the law of the cosines, such angles

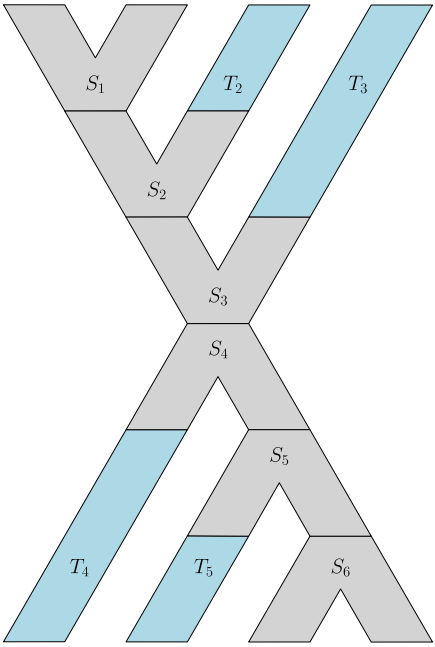

Fig. 9. Schema of the variable gadget when the maximum degree δ_ϕ of G_ϕ is 4.

are equal to $\arccos\left(\frac{w_1^2+w_3^2-w_2^2}{2w_1w_3}\right) > \arccos(0.57) > 54°$ and $\arccos\left(\frac{w_1}{2w_2}\right) > \arccos(0.56) > 55°$, respectively. □

Using Property 2, the proof of Lemma 4 can be easily adapted to obtain the following.

Lemma 5. *In any planar straight-line realization of the split gadget, if $\triangle_{IN}$ is drawn inside $\triangle_s$, then $\triangle^1_{OUT}$ is drawn outside $\triangle^1_t$ and $\triangle^2_{OUT}$ is drawn outside $\triangle^2_t$. Further, if $\triangle^1_{OUT}$ is drawn inside $\triangle^1_t$ or $\triangle^2_{OUT}$ is drawn inside $\triangle^2_t$, then $\triangle_{IN}$ is drawn outside $\triangle_s$.*

The variable gadget. Remember that δ_ϕ denotes the maximum degree of G_ϕ. Let $S_1, S_2, \ldots, S_{2\delta_\phi-2}$ be $2\delta_\phi - 2$ split gadgets, and $T_2, T_3, \ldots, T_{2\delta_\phi-3}$ be $2\delta_\phi - 4$ straight transmission gadgets such that, for $i = 2, \ldots, \delta_\phi - 1$, the gadgets T_i and $T_{2\delta_\phi-i-1}$ are $4(i-1)$-transmission gadgets. The variable gadget can be obtained by combining these gadgets as follows; refer to Fig. 9.

- For $i = 1, \ldots, \delta_\phi - 2$, we combine S_i and S_{i+1} by identifying the attachment edge e_{IN} of S_i with the attachment edge e^1_{OUT} of S_{i+1};
- we combine $S_{\delta_\phi-1}$ and S_{δ_ϕ} by identifying the attachment edge e_{IN} of $S_{\delta_\phi-1}$ with the attachment edge e_{IN} of S_{δ_ϕ};
- for $i = \delta_\phi, \ldots, 2\delta_\phi - 3$, we combine S_i and S_{i+1} by identifying the attachment edge e^1_{OUT} of S_i with the attachment edge e_{IN} of S_{i+1}; and
- for $i = 2, \ldots, 2\delta_\phi - 3$, we combine S_i and T_i by identifying the attachment edge e^2_{OUT} of S_i with the attachment edge e_{IN} of T_i.

By Observation 9 and the fact that transmission and split gadgets are 2-trees, we have that the variable gadget is a 2-tree and its frame is a maximal outerplanar graph.

The variable gadget provides $2\delta_\phi$ attachment edges; refer to Fig. 10. Namely:

- the attachment edges e^1_{OUT} and e^2_{OUT} of S_1,

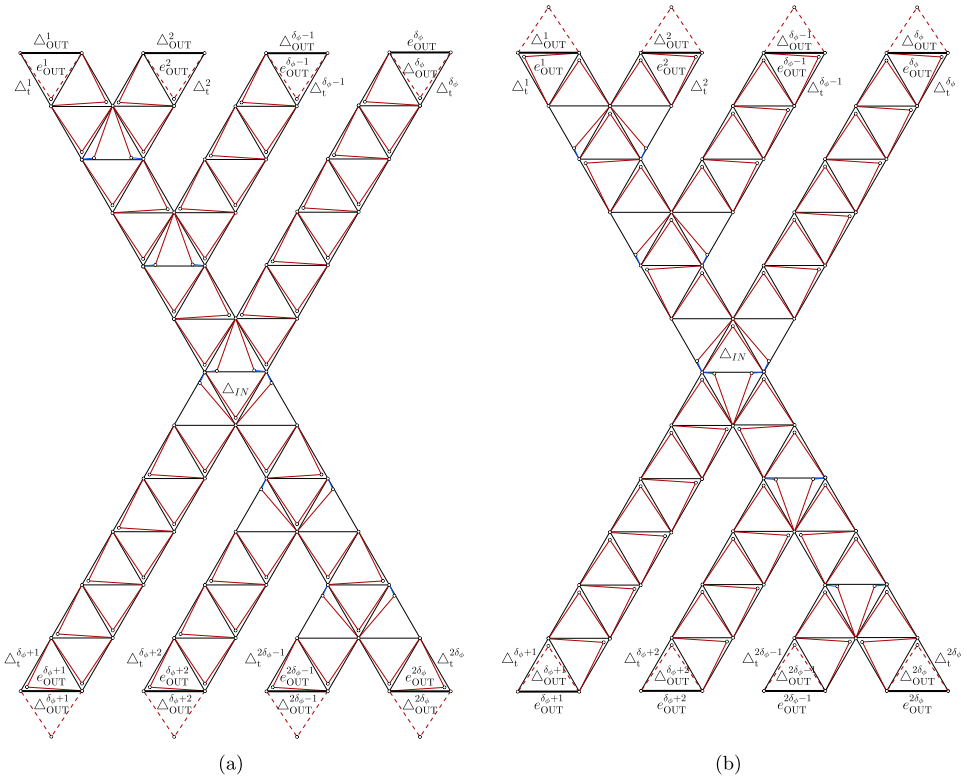

Fig. 10. The variable gadget when the maximum degree δ_ϕ of G_ϕ is 4. In the logic of the reduction, the realization on the left corresponds to an assignment of `True` to the variable represented by the variable gadget, while the realization on the right corresponds to an assignment of `False` to the variable represented by the variable gadget.

- for $i = 2, \ldots, 2\delta_\phi - 3$, the attachment edge e_{OUT} of T_i, which we denote as e^{i+1}_{OUT}, and
- the attachment edges e^1_{OUT} and e^2_{OUT} of $S_{2\delta_\phi-2}$, which we denote as $e^{2\delta_\phi-1}_{\text{OUT}}$ and $e^{2\delta_\phi}_{\text{OUT}}$, respectively.

For $i = 1, \ldots, 2\delta_\phi$, we also denote by $\triangle^i_{\text{OUT}}$ and by $\triangle^i_t$ the transmission triangle and the frame triangle incident to e^i_{OUT}, respectively. Finally, let $\triangle_{\text{IN}}$ be the transmission triangle that is incident to the attachment edge shared by the split gadgets $S_{\delta_\phi-1}$ and S_{δ_ϕ}. We call $\triangle_{\text{IN}}$ the *truth-assignment triangle*. Intuitively, the truth value associated with a planar straight-line realization of the variable gadget, will depend on whether $\triangle_{\text{IN}}$ lies inside the frame triangle $\triangle_s$ of $S_{\delta_\phi-1}$ or inside the frame triangle $\triangle_s$ of S_{δ_ϕ} in such a realization. We show the following simple and yet useful geometric property of the variable gadget.

Property 3. *In any planar straight-line realization of the variable gadget, up to a rotation, the frame lies in an axis-aligned box B of width $w_1(2\delta_\phi - 1)$ and height $2\sqrt{3}w_1(\delta_\phi - 1)$. Furthermore, the attachment edges $e^1_{OUT}, \ldots, e^{\delta_\phi}_{OUT}$ lie along the top side of B at distance w_1 from each other, and the attachment edges $e^{\delta_\phi+1}_{OUT}, \ldots, e^{2\delta_\phi}_{OUT}$ lie along the bottom side of B at distance w_1 from each other.*

We have the following lemma. Refer to Fig. 10.

Lemma 6. *In any planar straight-line realization of the variable gadget, if the truth-assignment triangle $\triangle_{IN}$ is drawn inside the frame triangle $\triangle_s$ of $S_{\delta_\phi-1}$ (resp. of S_{δ_ϕ}), then, for $i = 1, \ldots, \delta_\phi$ (resp. for*

$i = \delta_\phi+1, \ldots, 2\delta_\phi$), the transmission triangle $\triangle^i_{OUT}$ is drawn outside the frame triangle $\triangle^i_t$. Furthermore, if there is some $i \in \{1, \ldots, \delta_\phi\}$ (resp. some $i \in \{\delta_\phi + 1, \ldots, 2\delta_\phi\}$) for which the transmission triangle $\triangle^i_{OUT}$ is drawn inside the frame triangle $\triangle^i_t$, then $\triangle_{IN}$ is drawn inside the frame triangle $\triangle_s$ of S_{δ_ϕ} (resp. of $S_{\delta_\phi-1}$).

Proof. We start with the first part of the statement. Let Γ be a planar straight-line realization of the variable gadget. Suppose the truth-assignment triangle $\triangle_{\text{IN}}$ is drawn inside the frame triangle $\triangle_s$ of $S_{\delta_\phi-1}$ in Γ. We show that, for $i = 1, \ldots, \delta_\phi$, the transmission triangle $\triangle^i_{\text{OUT}}$ is drawn outside the frame triangle $\triangle^i_t$ in Γ. By a symmetric argument we can proof that, if $\triangle_{\text{IN}}$ is drawn inside the frame triangle $\triangle_s$ of S_{δ_ϕ} in Γ, then, for $i = \delta_\phi, \ldots, 2\delta_\phi$, the transmission triangle $\triangle^i_{\text{OUT}}$ is drawn outside the frame triangle $\triangle^i_t$ in Γ. We have the following claims.

Claim 1. *Consider any two split gadgets S_i and S_{i-1}, for $i = \delta_\phi - 1, \ldots, 2$. If the transmission triangle $\triangle^1_{OUT}$ of S_i is drawn outside the frame triangle $\triangle^1_t$ of S_i, then the transmission triangle $\triangle^1_{OUT}$ of S_{i-1} is drawn outside the frame triangle $\triangle^1_t$ of S_{i-1}.*

Proof. Observe that, if the transmission triangle $\triangle^1_{\text{OUT}}$ of S_i is drawn outside the frame triangle $\triangle^1_t$ of S_i, then it lies inside the frame triangle $\triangle_s$ of S_{i-1}. Since the transmission triangle $\triangle^1_{\text{OUT}}$ is in fact a copy of the transmission triangle $\triangle_{IN}$ of S_{i-1} (which has been removed when combining S_i and S_{i-1}), by Lemma 5 we have that the transmission triangle $\triangle^1_{\text{OUT}}$ of S_{i-1} is drawn outside the frame triangle $\triangle^1_t$ of S_{i-1}. □

Claim 2. *Consider any split gadget S_i and any transmission gadget T_i, for $i = \delta_\phi - 1, \ldots, 2$. If the transmission triangle $\triangle^2_{OUT}$ of S_i is drawn outside the frame triangle $\triangle^2_t$ of S_i, then the transmission triangle $\triangle_{OUT}$ of T_i is drawn outside the frame triangle $\triangle_t$ of T_i.*

Proof. Observe that, if the transmission triangle $\triangle^2_{\text{OUT}}$ of S_i is drawn outside the frame triangle $\triangle^2_t$ of S_i, then it lies inside the frame triangle $\triangle_s$ of T_i. Since $\triangle^2_{\text{OUT}}$ is in fact a copy of the transmission triangle $\triangle_{IN}$ of T_i (which has been removed when combining S_i and T_i), by Lemma 4 we have that the transmission triangle $\triangle_{\text{OUT}}$ of T_i is drawn outside the frame triangle $\triangle_t$ of T_i. □

Since the truth-assignment triangle $\triangle_{\text{IN}}$ lies inside the frame triangle $\triangle_s$ of $S_{\delta_\phi-1}$, then by Lemma 5, the transmission triangles $\triangle^1_{\text{OUT}}$ and $\triangle^2_{\text{OUT}}$ of $S_{\delta_\phi-1}$ are drawn outside the frame triangles $\triangle^1_t$ and $\triangle^2_t$ of $S_{\delta_\phi-1}$, respectively. Hence, in the variable gadget, the transmission triangle $\triangle^i_{\text{OUT}}$ is drawn outside the frame triangle $\triangle^i_t$; this follows from Claim 1 for $i = 1, 2$, and from Claim 2 for $i = 3, \ldots, \delta_\phi$.

We now prove the second part of the statement. Let Γ be a planar straight-line realization of the variable gadget. Suppose that, for some $j \in \{1, \ldots, \delta_\phi\}$, there is a transmission triangle $\triangle^j_{\text{OUT}}$ of the variable gadget drawn inside the frame triangle $\triangle^j_t$ of the variable gadget. We show that the truth-assignment triangle $\triangle_{\text{IN}}$ is drawn inside the frame triangle $\triangle_s$ of S_{δ_ϕ} in Γ. By a symmetric argument we can proof that, if there is some $j \in \{\delta_\phi + 1, \ldots, 2\delta_\phi\}$ for which a transmission triangle $\triangle^j_{\text{OUT}}$ of the variable gadget is drawn inside the frame triangle $\triangle^j_t$ of the variable gadget, then the truth-assignment triangle $\triangle_{\text{IN}}$ is drawn inside the frame triangle $\triangle_s$ of $S_{\delta_\phi-1}$ in Γ. We have the following two claims, whose proofs are symmetric to the proofs of Claims 1 and 2.

Claim 3. *Consider any two split gadgets S_i and S_{i-1}, for $i = \delta_\phi - 1, \ldots, 2$. If the transmission triangle $\triangle^1_{OUT}$ of S_{i-1} is drawn inside the frame triangle $\triangle^1_t$ of S_{i-1}, then the transmission triangle $\triangle^1_{OUT}$ of S_i is drawn inside the frame triangle $\triangle^1_t$ of S_i.*

Claim 4. *Consider any split gadget S_i and any transmission gadget T_i, for $i = \delta_\phi - 1, \ldots, 2$. If the transmission triangle $\triangle_{OUT}$ of T_i is drawn inside the frame triangle $\triangle_t$ of T_i, then the transmission triangle $\triangle^2_{OUT}$ of S_i is drawn inside the frame triangle $\triangle^2_t$ of S_i.*

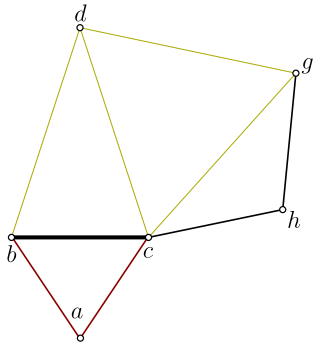

Fig. 11. The triangles (a, b, c), (d, b, c), (g, d, c), and (g, h, c) of the flag gadget.

If $1 \leq j \leq 2$, then the transmission triangle $\triangle^j_{\text{OUT}}$ of the variable gadget is actually a transmission triangle of the split gadget S_1. Hence, by Claim 3, we have that the transmission triangle $\triangle^1_{\text{OUT}}$ of $S_{\delta_\phi-1}$ is drawn inside the frame triangle $\triangle^1_t$ of $S_{\delta_\phi-1}$. If instead $3 \leq j \leq \delta_\phi$, then the transmission triangle $\triangle^j_{\text{OUT}}$ of the variable gadget is actually a transmission triangle of a transmission gadget. Hence, by Claim 4, we have that the transmission triangle $\triangle^2_{\text{OUT}}$ of S_j is drawn inside the frame triangle $\triangle^2_t$ of S_j. This fact and again Claim 3 imply that either the transmission triangle $\triangle^1_{\text{OUT}}$ of $S_{\delta_\phi-1}$ is drawn inside the frame triangle $\triangle^1_t$ of $S_{\delta_\phi-1}$ (which happens when $j < \delta_\phi$), or the transmission triangle $\triangle^2_{\text{OUT}}$ of $S_{\delta_\phi-1}$ is drawn inside the frame triangle $\triangle^2_t$ of $S_{\delta_\phi-1}$ (which happens when $j = \delta_\phi$). In all cases, by Lemma 5, the truth-assignment triangle $\triangle_{\text{IN}}$ is drawn inside the frame triangle $\triangle_s$ of S_{δ_ϕ} in Γ. □

The flag gadget. The flag gadget consists of the following triangles (see Figs. 11 and 12):

(i) a transmission triangle (a, b, c) with base (b, c),
(ii) a tall isosceles triangle (d, b, c) with base (b, c) of length w_1 and two longer sides of length w_4,
(iii) a flat isosceles triangle (f, d, c) with base (d, c) of length w_4 and two shorter sides of length w_2,
(iv) an equilateral triangle (g, d, c) with sides of length w_4,
(v) a flat isosceles triangle (h, g, c) with base (g, c) of length w_4 and two shorter sides of length w_1,
(vi) four tall isosceles triangles (h, i, g), (i, l, g), (h, m, c), and (m, n, c) with bases (h, i), (i, l), (h, m), and (m, n), respectively, of length w_3 and two longer sides of length w_1.

By Observation 9 and the fact that any two of the listed triangles share at most one edge, we have that the flag gadget is a 2-tree. The flag gadget provides a single attachment edge, which coincides with the base (b, c) of the transmission triangle (a, b, c).

We will exploit the following geometric properties of the flag gadget.

Property 4. *The flag gadget admits, among others, planar straight-line realizations in which:*

(a) The triangle (a, b, c) lies inside the triangle (d, b, c), and the triangles (h, m, c) and (m, n, c) lie inside the triangle (h, g, c); see Fig. 12(a).
(b) The triangle (a, b, c) lies inside the triangle (d, b, c), and the triangles (h, i, g) and (i, l, g) lie inside the triangle (h, g, c); see Fig. 12(b).
(c) The triangle (a, b, c) lies outside the triangle (d, b, c) and the triangles (h, m, c), (m, n, c), (h, m, c), and (m, n, c) lie inside the triangle (h, g, c); see Fig. 12(c).

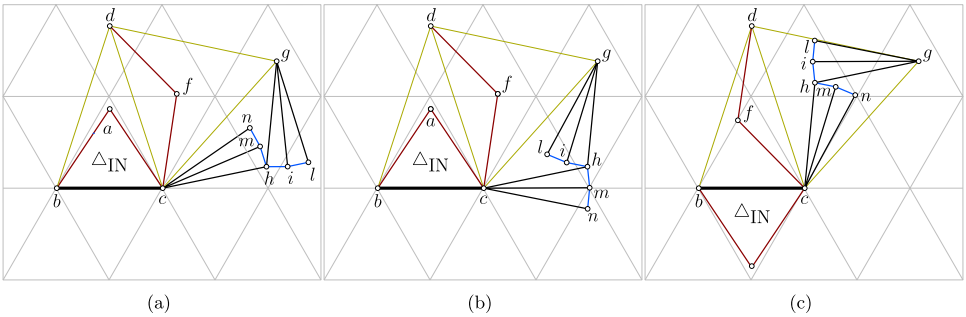

Fig. 12. The flag gadget and three planar straight-line realizations of it. In (a) and (b) the transmission triangle (a, b, c) lies inside the triangle (d, b, c), while in (c) it lies outside (d, b, c). In (a) the isosceles triangles (h, m, c) and (m, n, c) lie inside (h, g, c), in (b) the isosceles triangles (h, i, g) and (i, l, g) lie inside (h, g, c), and in (c) all four the isosceles triangles (h, m, c), (m, n, c), (h, m, c), and (m, n, c) lie inside (h, g, c).

Property 5. *The values of some relevant angles of the triangles forming the flag gadget are the following:*

- $\widehat{bcd} := \arccos \frac{w_1}{2w_4} = \arccos \frac{1}{3.22} \simeq 71.91°$
- $\widehat{bca} := \arccos \frac{w_1}{2w_2} = \arccos \frac{1}{1.8} \simeq 56.25°$
- $\widehat{fcd} := \arccos \frac{w_4}{2w_2} = \arccos \frac{1.61}{1.8} \simeq 26.56°$
- $\widehat{hcg} := \arccos \frac{w_4}{2w_1} = \arccos \frac{1.61}{2} \simeq 36.39°$
- $\widehat{chg} := 180° - 2 \cdot \widehat{hcg} \simeq 107.22°$
- $\widehat{chm} := \arccos \frac{w_3}{2w_1} = \arccos \frac{0.2}{2} \simeq 84.26°$
- *Let λ be the smallest internal angle of the tall isosceles triangles with two sides of length w_1 and one side of length w_3. We have $\lambda = 180° - 2 \cdot \widehat{chm} \simeq 180° - 2 \cdot 84.26° = 11.48°$.*

We show the following.

Lemma 7. *In any planar straight-line realization of the flag gadget, the following statements hold true:*

(a) If the transmission triangle (a, b, c) is drawn inside the triangle (d, b, c), then the triangle (h, g, c) is drawn outside the triangle (g, d, c).
(b) Not both the vertices l and n lie inside the triangle (h, g, c).

Proof. Consider a planar straight-line realization of the flag gadget. By Observations 1 and 2, in such a realization the triangles (d, b, c) and (g, d, c) are not drawn inside one another. Hence, the triangle (f, d, c) lies inside (d, b, c) if and only if it lies outside (g, d, c).

To prove statement (a), suppose the triangle (a, b, c) is drawn inside the triangle (d, b, c). By our discussion above and the fact that $\widehat{fcd} + \widehat{bca} > \widehat{bcd}$, then the triangle (f, d, c) lies inside (g, d, c). By this fact and the fact that $\widehat{fcd} + \widehat{hcg} > \widehat{dcg} = 60°$, we have that the triangle (h, g, c) is drawn outside (g, d, c). This ends the proof of statement (a).

To prove statement (b) it suffices to show that m and i cannot both lie inside the triangle (h, g, c). This follows from the fact that $\widehat{chm} = \widehat{ghi}$ and $2 \cdot \widehat{chm} > \widehat{chg}$. □

The clause gadget. We now describe the construction of the clause gadget. Such a construction is parametric with respect to the distances between the attachment edges of the transmission gadgets coming from the variable gadgets whose literals appear in the clause. By Property D7 and since the attachment edges of the transmission gadgets have length one, the horizontal distance between the two leftmost attachment edges is $\frac{\alpha}{2} + 2$, for some non-negative even integer value α, and the horizontal distance between the two rightmost attachment edges is $\frac{\beta}{2} + 3$, for some non-negative even integer value β. Based on the values α and β, we define the clause gadget, which we refer to

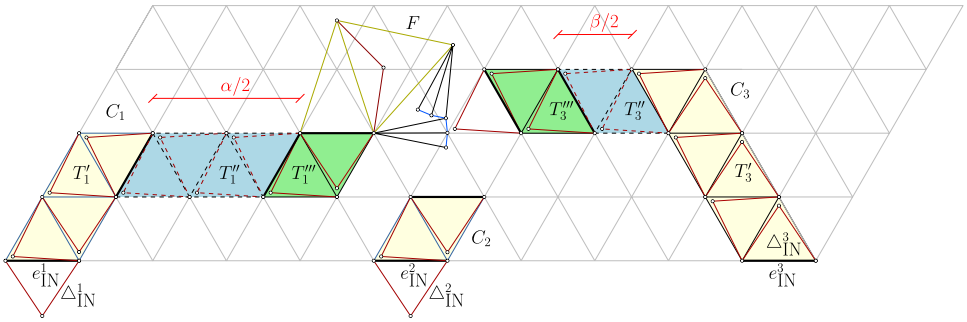

Fig. 13. The (α, β)-clause gadget with $\alpha = 4$ and $\beta = 2$.

as the (α, β)-*clause gadget*. The gadget provides three attachment edges e^1_{IN}, e^2_{IN}, and e^3_{IN} incident to the transmission triangles $\triangle^1_{\text{IN}}$, $\triangle^2_{\text{IN}}$, and $\triangle^3_{\text{IN}}$, respectively. The attachment edges e^1_{IN}, e^2_{IN}, and e^3_{IN} are identified with the attachment edges of the transmission gadgets coming from the variable gadgets whose literals appear in the clause. Also, recall that the transmission triangles $\triangle^1_{\text{IN}}$, $\triangle^2_{\text{IN}}$, and $\triangle^3_{\text{IN}}$ are identified with the ones of the transmission gadgets coming from the variable gadgets whose literals appear in the clause. In any planar straight-line realization of H_ϕ, the edges e^1_{IN}, e^2_{IN}, e^3_{IN} lie in this left-to-right order along the same horizontal line.

We now describe the (α, β)-clause gadget for a positive clause (the negative clause has a symmetric construction); refer to Fig. 13. We exploit the placement of e^1_{IN}, e^2_{IN}, e^3_{IN} (which is fixed, since such edges belong to the union of the frames of the variable, ladder, and transmission gadgets; see Section 4.2) as a reference for arranging the gadget components. The (α, β)-clause gadget consists of three subgraphs C_1, C_2, and C_3 defined as follows.

- The subgraph C_1 is obtained by combining into a single connected component a turning 4-transmission gadget T'_1, a straight α-transmission gadget T''_1, a turning 2-transmission gadget T'''_1, and a flag gadget F, as shown in Fig. 13. The attachment edge e_{IN} of T'_1 is the attachment edge e^1_{IN} of the clause gadget.
- The subgraph C_2 is a straight 2-transmission gadget, whose attachment edge e_{IN} is the attachment edge e^2_{IN} of the clause gadget.
- The subgraph C_3 is obtained by combining into a single connected component a turning 6-transmission gadget T'_3, a straight β-transmission gadget T''_3, and a straight 2-transmission gadget T'''_3, as depicted in Fig. 13. The attachment edge e_{IN} of T'_3 is the attachment edge e^3_{IN} of the clause gadget.

By Observation 9 and the fact that transmission and flag gadgets are 2-trees, we have that each of C_1, C_2, and C_3 is a 2-tree. We say that a straight-line realization of the (α, β)-clause gadget is *feasible*, if it satisfies the following properties:

(A) the attachment edges e^1_{IN}, e^2_{IN}, and e^3_{IN} lie in this left-to-right order on a horizontal line ℓ, with horizontal distance $\frac{\alpha}{2}+2$ between e^1_{IN} and e^2_{IN}, and horizontal distance $\frac{\beta}{2}+3$ between e^2_{IN} and e^3_{IN}; and
(B) the frame triangles of the clause gadget all lie above ℓ.

Consider any feasible planar straight-line realization of the (α, β)-clause gadget. In such realization, consider the trapezoid $\mathcal{T}$ whose base is the shortest horizontal segment containing e^1_{IN}, e^2_{IN}, and e^3_{IN}, whose lateral sides have slope 60°, and whose height is $2\sqrt{3}$. Observe that the base of $\mathcal{T}$ has length $8 + (\alpha + \beta)/2$. A key ingredient for the logic of the reduction is the following lemma.

Lemma 8. *The (α, β)-clause gadget admits a feasible planar straight-line realization if and only if at least one of the transmission triangles $\triangle^1_{IN}$, $\triangle^2_{IN}$, and $\triangle^3_{IN}$ lies outside $\mathcal{T}$.*

Proof. Consider any feasible planar straight-line realization of the (α, β)-clause gadget. Note that, for $i = 1, 2, 3$, if the transmission triangle $\triangle^i_{\text{IN}}$ lies inside $\mathcal{T}$, then it lies inside the frame triangle $\triangle_s$ of the transmission gadget composing C_i that contains the edge e^i_{IN}. By this observation and Lemma 4, we obtain the following properties:

Property 6. *Consider the transmission triangle (a, b, c) and the tall isosceles triangle (d, b, c) of the flag gadget F. If the transmission triangle $\triangle^1_{IN}$ lies inside $\mathcal{T}$, then the triangle (a, b, c) is drawn inside the triangle (d, b, c). Conversely, if the triangle (a, b, c) is drawn outside the triangle (d, b, c), then the transmission triangle $\triangle^1_{IN}$ lies outside $\mathcal{T}$.*

Property 7. *Consider the transmission triangle $\triangle_{OUT}$ and the frame triangle $\triangle_t$ of C_2. If the transmission triangle $\triangle^2_{IN}$ lies inside $\mathcal{T}$, then $\triangle_{OUT}$ lies outside $\triangle_t$. Conversely, if $\triangle_{OUT}$ lies inside $\triangle_t$, then the transmission triangle $\triangle^2_{IN}$ lies outside $\mathcal{T}$.*

Property 8. *Consider the transmission triangle $\triangle_{OUT}$ and the frame triangle $\triangle_t$ of T'''_3. If the transmission triangle $\triangle^3_{IN}$ lies inside $\mathcal{T}$, then $\triangle_{OUT}$ is drawn outside $\triangle_t$. Conversely, if $\triangle_{OUT}$ is drawn inside $\triangle_t$, then the transmission triangle $\triangle^3_{IN}$ lies outside $\mathcal{T}$.*

Suppose first that at least one of $\triangle^1_{\text{IN}}$, $\triangle^2_{\text{IN}}$, and $\triangle^3_{\text{IN}}$ lie outside $\mathcal{T}$. We prove that the (α, β)-clause gadget admits a feasible planar straight-line realization. We have the following cases:

- The transmission triangle $\triangle^1_{\text{IN}}$ lies outside $\mathcal{T}$.
  In this case the transmission triangles of T'_1, T''_1, and T'''_1 can be drawn in such a way that, in the flag gadget F, the transmission triangle (a, b, c) lies outside the triangle (b, c, d). This allows the flag gadget to adopt the realization (c) of Property 4 (see also Fig. 12(c)). Observe that, regardless of whether $\triangle^2_{\text{IN}}$ and $\triangle^3_{\text{IN}}$ lie inside $\mathcal{T}$, the transmission triangles of C_2, T'_3, T''_3, and T'''_3 can be drawn without crossings.
- The transmission triangle $\triangle^2_{\text{IN}}$ lies outside $\mathcal{T}$.
  In this case the transmission triangles of C_2 can be drawn in such a way that the transmission triangle $\triangle_{\text{OUT}}$ of C_2 lies inside the frame triangle $\triangle_t$ of C_2. This allows the flag gadget to adopt the realization (b) of Property 4 (see also Fig. 12(b)). Observe that, regardless of whether $\triangle^1_{\text{IN}}$ and $\triangle^3_{\text{IN}}$ lie inside $\mathcal{T}$, the transmission triangles of T'_1, T''_1, T'''_1, T'_3, T''_3, and T'''_3 can be drawn without crossings.
- The transmission $\triangle^3_{\text{IN}}$ lies outside $\mathcal{T}$.
  In this case the transmission triangles of T'_3, T''_3, and T'''_3 in such a way that the transmission triangle $\triangle_{\text{OUT}}$ of T'''_3 lies inside the frame $\triangle_t$ of T'''_3. This allows the flag gadget to adopt the realization (a) of Property 4 (see also Fig. 12(a)). Observe that, regardless of whether $\triangle^1_{\text{IN}}$ and $\triangle^2_{\text{IN}}$ lie inside $\mathcal{T}$, the transmission triangles of T'_1, T''_1, T'''_1, and C_2 can be drawn without crossings.

Suppose now that all $\triangle^1_{\text{IN}}$, $\triangle^2_{\text{IN}}$, and $\triangle^3_{\text{IN}}$ lie inside $\mathcal{T}$. We prove that the (α, β)-clause gadget does not admit a feasible planar straight-line realization. We analyze the possible planar straight-line realizations of the flag gadget F. By Property 6 and statement (a) of Lemma 7, the triangle (h, g, c) is drawn outside the triangle (g, d, c). On the other hand, by statement (b) of Lemma 7, we have that either (i) the vertex n lies outside the triangle (h, g, c) (refer to Fig. 14(a)), (ii) the vertex l lies outside the triangle (h, g, c) (refer to Fig. 14(b)), or (iii) both vertices n and l lie outside the triangle (h, g, c). We show next that in all cases, there is an intersection between the edges two triangles of the (α, β)-clause gadget.

Consider first the case in which the vertex n lies outside the triangle (h, g, c); refer to Fig. 14(a). The points o, p, q are the vertices of the transmission triangle $\triangle_{\text{OUT}}$ of C_2. We prove that $\widehat{con} + \widehat{qop} > 120°$ and that the length of the segment $\overline{on}$ is smaller than w_2. These two statements imply that the segment $\overline{cn}$ intersects the segment $\overline{oq}$. To prove both statements we consider an auxiliary isosceles triangle with vertices n, c, o (note that only the side $\overline{cn}$ of the triangle (n, c, o) corresponds to an edge of H_ϕ). We have that:

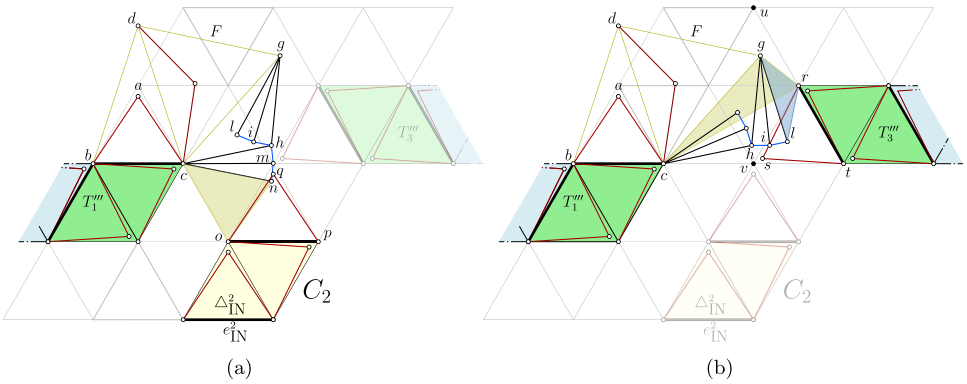

Fig. 14. Illustration for the proof of Lemma 8 when $\triangle^1_{\text{IN}}$, $\triangle^2_{\text{IN}}$, and $\triangle^3_{\text{IN}}$ lie all in the interior of $\mathcal{T}$.

- The internal angle at c is $\widehat{nco} \simeq 48.74°$. Namely, $\widehat{nco} := 240° - \widehat{bcd} - \widehat{dcg} - \widehat{gch} - 2\lambda$. By Property 5, we obtain $\widehat{nco} \simeq 240° - 71.91° - 60° - 36.39° - 2 \cdot 11.48° = 48.74°$.
- The internal angle at o is $\widehat{con} \simeq (180° - \widehat{nco})/2 = 65.63°$.
- The segments $\overline{co}$ and $\overline{cn}$ have length w_1.
- The segment $\overline{on}$ has length $2w_1 \sin(\frac{\widehat{nco}}{2}) \simeq 0.83$.

Since the triangle (q, o, p) is a transmission triangle, we have that $\widehat{qop} = \arccos(\frac{w_1}{2w_2}) = \arccos(\frac{1}{1.8}) = 56.25°$, and the segment $\overline{oq}$ has length w_2. Hence, $\widehat{con} + \widehat{qop} \simeq 65.63° + 56.25° = 121.88° > 120°$, and $|\overline{oq}| = w_2 = 0.9 > |\overline{on}| \simeq 0.83$. This implies that $\overline{cn}$ intersects $\overline{oq}$, as claimed.

Consider now the case in which the vertex l lies outside the triangle (h, g, c); refer to Fig. 14(b). The points r, t, s denote the vertices of the transmission triangle $\triangle_{\text{OUT}}$ of T_3'''. Let u be the point at distance $w_1 = 1$ from r such that the ray from r through u has slope 120°. We show that $\widehat{gru} + \widehat{lrg} + \widehat{trs} > 180°$ and that the length of the segment $\overline{rl}$ is smaller than w_2. These two statements imply that the segment $\overline{gl}$ intersects the segment $\overline{rs}$. To prove both statements we consider two auxiliary isosceles triangles. The first triangle has vertices c, g, and r. We have that:

- The segment $\overline{cg}$ has length w_4.
- The segment $\overline{cr}$ has length $|\overline{cr}| = 2\frac{\sqrt{3}}{2} = \sqrt{3}$.
- The internal angle at c is $\widehat{gcr} \simeq 18.09°$. Namely, $\widehat{bcr} = 150°$. Further, $\widehat{bcg} = \widehat{bcd} + \widehat{dcg}$ and, by Property 5, we have $\widehat{bcg} \simeq 71.91° + 60° = 131.91°$, hence $\widehat{gcr} = \widehat{bcr} - \widehat{bcg} \simeq 18.09°$.
- The segment $\overline{gr}$ has length $|\overline{gr}| \simeq 0.54$. Namely, by the law of cosines, we have $|\overline{gr}|^2 = |\overline{cg}|^2 + |\overline{cr}|^2 - 2|\overline{cg}| \cdot |\overline{cr}| \cdot \cos(\widehat{gcr})$. Since $|\overline{cg}| = w_4 = 1.61$ and $|\overline{cr}| = \sqrt{3}$, we obtain $|\overline{gr}|^2 \simeq 2.5921 + 3 - 2 \cdot 1.61 \cdot \sqrt{3} \cdot \cos(18.09°) \simeq 0.29$, hence $|\overline{gr}| \simeq 0.54$.
- The internal angle at r is $\widehat{crg} \simeq 68.25°$. Namely, by the law of cosines, we have $\widehat{crg} = \arccos\left(\frac{|\overline{gr}|^2 + |\overline{cr}|^2 - |\overline{gc}|^2}{2|\overline{gr}| \cdot |\overline{cr}|}\right) \simeq \arccos\left(\frac{0.54^2 + 3 - 1.61^2}{2 \cdot 0.54 \cdot \sqrt{3}}\right) \simeq \arccos\left(\frac{0.6924}{1.8684}\right) \simeq \arccos(0.3706) \simeq 68.25°$.
- The internal angle at g is $\widehat{rgc} = 180° - \widehat{crg} - \widehat{gcr} \simeq 180° - 68.25° - 18.09° = 93.66°$.

The second auxiliary triangle we consider has vertices g, r, and l. We have that:

- The segment $\overline{lg}$ has length $w_1 = 1$.
- The internal angle at g is $\widehat{rgl} \simeq 34.31°$. Namely, $\widehat{lgc} = \widehat{hgc} + 2\lambda$. By Property 5, we have $\widehat{lgc} \simeq 36.39° + 2 \cdot 11.48° = 59.35°$. We obtain that $\widehat{rgl} = \widehat{rgc} - \widehat{lgc} \simeq 93.66° - 59.35° = 34.31°$.
- The edge $\overline{lr}$ has length $|\overline{lr}| \simeq 0.63$. Namely, by the law of cosines, we have that $|\overline{lr}|^2 = |\overline{lg}|^2 + |\overline{rg}|^2 - 2|\overline{lg}| \cdot |\overline{rg}| \cos(\widehat{rgl})$, and hence $|\overline{lr}| \simeq \sqrt{1 + 0.54^2 - 2 \cdot 0.54 \cdot \cos(34.31°)} \simeq \sqrt{0.3995} \simeq 0.63$.

– The internal angle at r is $\widehat{lrg} = 117.2°$. Namely, by the law of cosines, we have $\widehat{lrg} = \arccos\left(\frac{|\overline{gr}|^2+|\overline{lr}|^2-|\overline{gl}|^2}{2|\overline{gr}|\cdot|\overline{lr}|}\right) \simeq \arccos\left(\frac{0.54^2+0.63^2-1^2}{2\cdot 0.54\cdot 0.63}\right) = \arccos\left(\frac{-0.3115}{0.6804}\right) \simeq \arccos(-0.4578) \simeq 117.2°$.

Let v be the point at distance $w_1 = 1$ from r such that the ray from r through v has slope 240°. Note that $\widehat{vru} = \widehat{vrc} + \widehat{crg} + \widehat{gru}$, hence $\widehat{gru} = \widehat{vru} - \widehat{vrc} - \widehat{crg} \simeq 120° - 30° - 68.25° = 21.75°$. On the other hand, since the triangle (r, s, t) is a transmission triangle, we have $\widehat{trs} \simeq 56.25°$ and the segment $\overline{rs}$ has length w_2. Hence $\widehat{gru} + \widehat{lrg} + \widehat{trs} \simeq 21.75° + 117.2° + 56.25° = 195.2° > 180°$, and $|\overline{rl}| \simeq 0.63 < w_2 = 0.9$. This implies that $\overline{gl}$ intersects $\overline{rs}$, as claimed, and concludes the proof of the lemma. □

4.4. Proof of the reduction

We are now ready to prove the main result of this section.

Proof of Theorem 8. We first prove that, starting from the boolean formula ϕ, the incidence graph G_ϕ of ϕ, and the monotone rectilinear representation Γ_ϕ of G_ϕ, we can construct the 2-tree H_ϕ in polynomial time. The first step of the construction is to obtain from Γ_ϕ the auxiliary drawing Γ_ϕ^* we described in Section 4.1. By Lemma 3, we can construct Γ_ϕ^* in polynomial time. The next step is to construct H_ϕ using Γ_ϕ^* as an auxiliary tool. We obtain H_ϕ by introducing a variable gadget, a clause gadget, and a transmission gadget for each variable, clause, and edge in G_ϕ, respectively, and then combining these gadgets together by identifying their attachment edges. In particular, we exploit Γ_ϕ^* to (i) define the size k of each k-transmission gadget that represents an edge of G_ϕ, (ii) define the parameters α and β of each (α, β)-clause gadget, and (iii) select the appropriate attachment edges to combine the gadgets together. We then merge the frames of the variable gadgets that are consecutive in the left-to-right order of the rectangles representing variables in Γ_ϕ^*. This is done by means of *ladder gadgets* (the shaded blue triangles in Fig. 6), which are maximal outerpaths, each composed of a sequence of frame triangles. We have that H_ϕ is a 2-tree by repeated applications of Observation 9, and the union of the frame triangles induces a maximal outerplanar graph. Moreover, from the detailed description of the gadgets construction of Section 4.3, it is not hard to see that each edge of H_ϕ has one of four possible weights, and that H_ϕ can be constructed in polynomial time.

We now prove that H_ϕ admits a planar straight-line realization *if and only if* ϕ is satisfiable. Suppose first that H_ϕ admits a planar straight-line realization Γ_H. We show that ϕ is satisfiable. For each variable v of ϕ, consider the variable gadget $\mathcal{V}$ of H_ϕ modeling v, the truth-assignment triangle $\triangle_{\text{IN}}$ of $\mathcal{V}$, and the frame triangle $\triangle_s$ of the split gadget S_{δ_ϕ} of $\mathcal{V}$; refer to Figs. 9 and 10. We set $v = \texttt{True}$ if and only if $\triangle_{\text{IN}}$ is drawn inside $\triangle_s$. We next prove this truth assignment satisfies every clause c of ϕ. Assume that c is a positive clause with variables v_1, v_2, and v_3, that is, $c = (v_1 \vee v_2 \vee v_3)$. The proof for the case in which c is negative is symmetric. Let $\mathcal{C}$ denote the clause gadget modeling c and, for $i = 1, 2, 3$, let $\mathcal{T}_i$ denote the transmission gadget modeling the edge (v_i, c) of G_ϕ. From the construction of the clause gadget, the realization of $\mathcal{C}$ in Γ_H is such that the frame of the components of $\mathcal{C}$ are bounded by a trapezoid $\mathcal{T}$. By Lemma 8, at least one of the triangles $\triangle^1_{\text{IN}}$, $\triangle^2_{\text{IN}}$, and $\triangle^3_{\text{IN}}$ of $\mathcal{C}$ lies outside $\mathcal{T}$. We assume w.l.o.g. that it is actually $\triangle^1_{\text{IN}}$ the triangle that lies outside $\mathcal{T}$. Then $\triangle^1_{\text{IN}}$ lies inside the frame triangle $\triangle_t$ of the transmission gadget $\mathcal{T}_1$. Let $\mathcal{V}_1$ denote the variable gadget modeling the variable v_1. By Lemma 4, there exists an index $j \in \{1, 2, \ldots, \delta_\phi\}$ such that the transmission triangle $\triangle^j_{\text{OUT}}$ of $\mathcal{V}_1$ lies inside the frame triangle $\triangle^j_t$ of $\mathcal{V}_1$ (for such index j the transmission triangle $\triangle^j_{\text{OUT}}$ is shared by $\mathcal{V}_1$ and $\mathcal{T}_1$). Thus, by Lemma 6, the truth-assignment triangle $\triangle_{\text{IN}}$ of $\mathcal{V}_1$ is drawn inside the frame triangle $\triangle_s$ of the split gadget S_{δ_ϕ} of $\mathcal{V}_1$. This implies in turn that v_1 is assigned the value $\texttt{True}$. Since this variable appears as a positive literal in c, we have that c is satisfied. This shows that the constructed truth assignment satisfies all the clauses of ϕ.

Suppose now that ϕ is satisfiable. We show that H_ϕ admits a planar straight-line realization Γ_H. Let τ be a satisfying truth assignment for ϕ. For each variable v of ϕ, let $\tau(v)$ denote the truth value of v in τ. Observe that, up to a rigid transformation, the union of all the frame triangles of

H_ϕ admits a unique planar straight-line realization. We initialize Γ_H to such a realization. Further, for each variable v, we adopt the configuration of Fig. 10(a) if $\tau(v) = \texttt{True}$ and the configuration of Fig. 10(b) if $\tau(v) = \texttt{False}$. For each transmission gadget modeling an edge of G_ϕ incident to a variable v and to a positive clause c, we adopt the configuration of Fig. 7(a)(right) if $\tau(v) = \texttt{True}$ and the configuration of Fig. 7(a)(left) if $\tau(v) = \texttt{False}$; a symmetric choice is made if c is negative. Since, for each clause c of ϕ, there exists at least one literal that is True, the triangle $\triangle^i_{\text{IN}}$, $i \in \{1, 2, 3\}$, associated with this literal is drawn outside the trapezoid $\mathcal{T}$ that bounds the frame of the components of $\mathcal{C}$ in Γ_H. Then, by Lemma 8, the clause gadget modeling c admits a feasible planar straight-line realization; such realizations are used to complete the planar straight-line realization Γ_H of H_ϕ. This concludes the proof of Theorem 8.

5. A linear-time algorithm for 2-trees with two edge lengths

In this section, we study the FEPR problem for weighted 2-trees in which each edge can only have one of at most two distinct lengths. We prove that, in this case, the FEPR problem is linear-time solvable. We first solve the case in which all the edges are prescribed to have the same length; we remark that the FEPR problem is NP-hard for general weighted planar graphs in which all the edges have the same length [28].

Theorem 10. *Let $G = (V, E, \lambda)$ be an n-vertex weighted 2-tree, where $\lambda : E \to \{w\}$ with $w \in \mathbb{R}^+$. There exists an $O(n)$-time algorithm that tests whether G admits a planar straight-line realization and, in the positive case, constructs such a realization.*

Proof. Suppose that there exists a planar straight-line realization Γ of G. Since all the edges of G have length w, every 3-cycle of G is represented in Γ by an equilateral triangle of side w, hence no triangle can be contained inside any other triangle in Γ. This, together with the fact that G is a 2-tree, implies that Γ is an outerplanar drawing of G. Thus, if G admits a planar straight-line realization, then it is a maximal outerplanar graph and, as such, it has a unique outerplane embedding [43,51].

Therefore, in order to test whether G has a planar straight-line realization, we test whether it is an outerplanar graph; this can be done in $O(n)$ time [25,42,54]. If the test fails, we conclude that G admits no planar straight-line realization. Otherwise, we construct in $O(n)$ time its unique outerplane embedding [25,42,43,51,54]. Finally, by means of Theorem 4, we test in $O(n)$ time whether G has a planar straight-line realization that respects the computed outerplane embedding. If the test fails, we conclude that G admits no planar straight-line realization. Otherwise, Theorem 4 provides us with the desired planar straight-line realization of G. □

We now extend our study to weighted graphs in which each edge is assigned with one of two possible lengths. We have the following main theorem.

Theorem 11. *Let $G = (V, E, \lambda)$ be an n-vertex weighted 2-tree, where $\lambda : E \to \{w_1, w_2\}$ with $w_1, w_2 \in \mathbb{R}^+$. There exists an $O(n)$-time algorithm that tests whether G admits a planar straight-line realization and, in the positive case, constructs such a realization.*

In the remainder of the section, we prove Theorem 11. Hereafter, we assume, w.l.o.g., that $w_1 < w_2$. Also, for the sake of simplicity of the description, we assume that $w_1 = 1$. This is not a loss of generality, as any planar straight-line realization with two edge lengths $w_1, w_2 \in \mathbb{R}^+$ with $w_1 < w_2$ can be turned into one in which the smaller length is 1 (and the larger length is w_2/w_1) by scaling the realization up by $1/w_1$.

We note that the realization of any 3-cycle of G is one of the following types of triangles (refer to Fig. 15):

(i) an equilateral triangle of side 1 (a *small equilateral triangle*),
(ii) an equilateral triangle of side w_2 (a *big equilateral triangle*),
(iii) an isosceles triangle with base 1 and two sides of length w_2 (a *tall isosceles triangle*), and
(iv) an isosceles triangle with base w_2 and two sides of length 1 (a *flat isosceles triangle*).

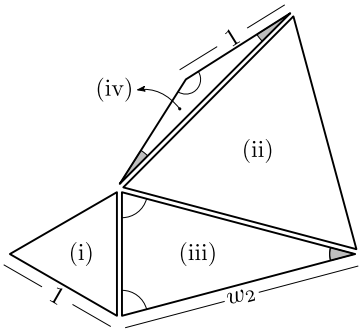

Fig. 15. The four interesting triangles. Since $w_2 > 1$, the angles in gray are always less than 60°, while the angles in white are always greater than 60°.

Any triangle of one of the types above is called an *interesting triangle*.

While small equilateral, big equilateral and tall isosceles triangle exist regardless of the value of w_2, the triangle inequality implies that flat isosceles triangles do not exist under the condition described in the following property.

Property 9. *If $w_2 \geq 2$, then no 3-cycle of G is realized as a flat isosceles triangle.*

5.1. Containment among triangles

The first central ingredient of our algorithm are the conditions satisfied by interesting triangles when they are drawn inside each other in a planar straight-line realization of G.

Lemma 9. *The interesting triangles satisfy the two properties below.*

(a) No interesting triangle can be drawn inside a small equilateral or a flat isosceles triangle.
(b) Neither a big equilateral nor a tall isosceles triangle can be drawn inside any interesting triangle.

Proof. First note that, from Observation 2, we can derive the following properties:

i. no interesting triangle can be drawn inside a small equilateral triangle;
ii. a big equilateral triangle cannot be drawn inside any interesting triangle; and
iii. no interesting triangle can be drawn inside an interesting triangle of the same type.

Furthermore, by Heron's formula, the area of a tall isosceles triangle is larger than the area of a flat isosceles triangle. Hence, again by Observation 2, we have the following property:

iv. a tall isosceles triangle cannot be drawn inside a flat isosceles triangle.

We now prove statement (a). The fact that no interesting triangle can be drawn inside a small equilateral triangle follows from Property i. To prove that an interesting triangle $\triangle_1$ cannot be drawn inside a flat isosceles triangle $\triangle_2$, we make a case analysis based on the type of interesting triangle of $\triangle_1$. The cases in which $\triangle_1$ is either a big equilateral triangle, a flat isosceles triangle, or a tall isosceles triangle, follow respectively from Properties Properties ii to iv. Consider now the case in which $\triangle_1$ is a small equilateral triangle. If $\triangle_1$ and $\triangle_2$ share an edge, then $\triangle_1$ cannot be drawn inside $\triangle_2$ by Observation 3. Suppose instead that $\triangle_1$ does not share an edge with $\triangle_2$. Let a, b, c be the vertices of $\triangle_2$, where $\overline{bc}$ is the side of $\triangle_2$ with length w_2; refer to Fig. 16. Let d be the orthogonal projection of the vertex a on the segment $\overline{bc}$, let R_1 denote the closed region bounded by the triangle with vertices a, c, d, and let R_2 denote the closed region bounded by the triangle with vertices a, b, d. Since $\triangle_2$ is flat isosceles, by Property 9 we have $w_2 < 2$, hence the lengths of the segments $\overline{bd}$, $\overline{cd}$,

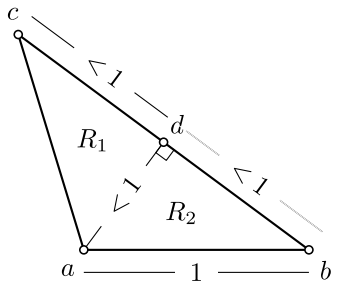

Fig. 16. Illustration for the proof of statement (a) of Lemma 9.

and $\overline{ad}$ are all smaller than 1. This implies that the only pair of points inside R_1 (resp. inside R_2) at distance greater than or equal to 1 is (a, c) (resp. (a, b)). Suppose, for a contradiction, that $\triangle_1$ is contained inside $\triangle_2$. Then two vertices of $\triangle_1$ are contained in the same triangle R_i, say they are contained in R_1. Since at least one of such two vertices does not coincide with a or c (otherwise $\triangle_1$ and $\triangle_2$ would share an edge), the distance between such two vertices is smaller than 1, while the length of the edge between them is at least 1, a contradiction. This ends the proof of statement (a).

We now prove statement (b). The fact that no big equilateral triangle can be drawn inside any interesting triangle follows from Property ii. To prove that a tall isosceles triangle $\triangle_1$ cannot be drawn inside an interesting triangle $\triangle_2$, we make a case analysis based on the type of interesting triangle of $\triangle_2$. The cases in which $\triangle_2$ is either a small equilateral triangle, a tall isosceles triangle, or a flat isosceles triangle, follow respectively from Properties i, iii and iv. The case in which $\triangle_2$ is a big equilateral triangle follows from Observation 1, given that $\triangle_1$ and $\triangle_2$ can share at most a single edge. This concludes the proof of statement (b) and the proof of the lemma. □

Let T denote the decomposition tree of G. Suppose that a planar straight-line realization Γ of G exists. The second central ingredient of our algorithm is the next lemma.

Lemma 10. *Let $\triangle_1$ and $\triangle_2$ be two triangles realizing two different 3-cycles of G in Γ. Suppose that $\triangle_1$ is drawn inside $\triangle_2$. Then $\triangle_1$ is a leaf triangle of T that shares a side with $\triangle_2$.*

Proof. By statement (a) of Lemma 9, we have that $\triangle_2$ is either big equilateral or tall isosceles. We discuss the two cases separately.

Suppose first that $\triangle_2$ is big equilateral. Consider any interesting triangle $\triangle_3$ that is drawn inside $\triangle_2$ in Γ and shares a side with $\triangle_2$. Note that $\triangle_3$ is flat isosceles; namely, $\triangle_3$ cannot be tall isosceles or big equilateral by statement (a) of Lemma 9, and it cannot be small equilateral since the sides of $\triangle_2$ have length w_2. Now suppose, for the sake of contradiction, that an interesting triangle $\triangle_4$ shares a side of length 1 with $\triangle_3$. Let v be the common vertex between $\triangle_2$, $\triangle_3$, and $\triangle_4$. If $\triangle_4$ is small equilateral or tall isosceles, as in Fig. 17(a) with $v = a$, then the sum of the angles at v of $\triangle_3$ and $\triangle_4$ exceeds 60°. Hence $\triangle_4$ crosses $\triangle_2$, a contradiction. Further, if $\triangle_4$ is flat isosceles, as in Fig. 17(a) with $v = b$, then $\triangle_4$ crosses $\triangle_2$ by Observation 1. It follows that the only interesting triangles inside $\triangle_2$ are flat isosceles triangles that are leaves of T and that share a side with $\triangle_2$.

Suppose next that $\triangle_2$ is tall isosceles. The proof follows similar arguments as for the case in which $\triangle_2$ is big equilateral, although, in this case, we distinguish two subcases. Namely, an interesting triangle $\triangle_3$ that is drawn inside $\triangle_2$ and that shares a side with $\triangle_2$ can be either small equilateral (and then it shares a side of length 1 with $\triangle_2$) as in Fig. 17(b), or flat isosceles (and then it shares a side of length w_2 with $\triangle_2$) as in Fig. 18(a).

Consider first the subcase in which $\triangle_3$ is small equilateral. Suppose, for the sake of contradiction, that an interesting triangle $\triangle_4$ shares a side of length 1 with $\triangle_3$. Let v be the common vertex between $\triangle_2$, $\triangle_3$, and $\triangle_4$. If $\triangle_4$ is small equilateral or tall isosceles, as in Fig. 17(b) with $v = a$, then

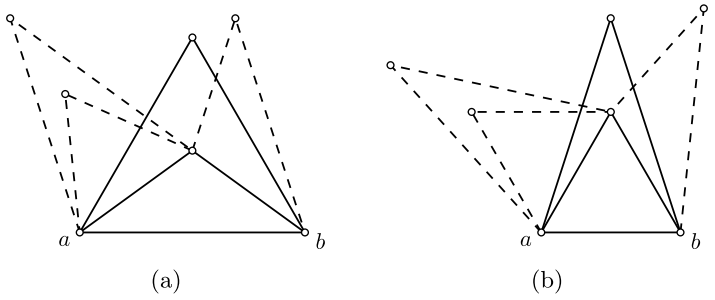

Fig. 17. Two illustrations for the proof of Lemma 10. (a) The case in which $\triangle_2$ is a big equilateral triangle and $\triangle_3$ is a flat isosceles triangle. (b) The case in which $\triangle_2$ is a tall isosceles triangle and $\triangle_3$ is a small equilateral triangle. In both figures, the possible cases for $\triangle_4$ are shown with dashed lines.

the sum of the angles at v of $\triangle_3$ and $\triangle_4$ exceeds 90°. Hence $\triangle_4$ crosses $\triangle_2$, a contradiction. On the other hand, if $\triangle_4$ is flat isosceles, as in Fig. 17(b) with $v = b$, then $\triangle_4$ crosses $\triangle_2$ by Observation 1.

Consider next the subcase in which $\triangle_3$ is flat isosceles. Suppose, for the sake of contradiction, that an interesting triangle $\triangle_4$ shares a side of length 1 with $\triangle_3$. Let v be the common vertex between $\triangle_2$, $\triangle_3$, and $\triangle_4$. Let a and b be the vertices incident to the base of $\triangle_2$, and let c be the third vertex of $\triangle_2$; refer to Fig. 18(a). The following cases arise depending of whether v is equal to c or a:

- Case 1: $v = c$. If $\triangle_4$ is small equilateral or tall isosceles, then the angle at v of $\triangle_4$ is larger than or equal to 60°, which is larger than the angle at v of $\triangle_2$. Hence $\triangle_4$ crosses $\triangle_2$. If $\triangle_4$ is instead flat isosceles, then $\triangle_4$ crosses $\triangle_2$ by Observation 1. In both case we have a contradiction.
- Case 2: $v = a$. If $\triangle_4$ is flat isosceles or tall isosceles, then $\triangle_4$ crosses $\triangle_2$ by Observation 1, a contradiction. Consider that $\triangle_4$ is instead small equilateral, as in Fig. 18(a). Let $\angle z$ denote the interior angle of $\triangle_2$ at vertex z, for $z \in \{a, b, c\}$. Let C be the circle with center on a and radius 1. The circle cuts the interior of the segment $\overline{ac}$ in a point y and it cuts the interior of the segment $\overline{bc}$ in a point x, since the length of $\overline{ac}$ is $w_2 > 1$ and since $\angle b < 90°$; see Fig. 18(b). If $\triangle_4$ is drawn inside $\triangle_2$, the vertices of $\triangle_4$ that are different from a lie on the circular arc of C that connects x and y through the interior of $\triangle_2$. Furthermore, since $\triangle_4$ is equilateral, the internal angle at a of the triangle with vertices a, c, x should be greater than 60°. Let θ denote such angle. We show that θ is actually less than 60°, which is a contradiction. Note that $\triangle_2$ and the triangle with vertices a, b, x are similar to each other, hence $\theta = \angle a - \angle c$. Since $\angle c + 2\angle a = 180°$ we have that

$$\theta = 3\angle a - 180°. \tag{1}$$

On the other hand, we have that $\cos(\angle a) = \frac{1}{2w_2}$ and, since $\triangle_1$ exists only if $\frac{1}{w_2} > \frac{1}{2}$, then $\cos(\angle a) > \frac{1}{4}$. Hence

$$\angle a < \arccos\left(\frac{1}{4}\right) \simeq 75.52°, \tag{2}$$

and from Eqs. (1) and (2) we obtain that $\theta = 3\angle a - 180° \simeq 46.56^o < 60^o$.

It follows that the only triangles inside $\triangle_2$ are flat isosceles triangles and small equilateral triangles that are leaves of T and that share a side with $\triangle_2$.

This concludes the proof of the lemma. □

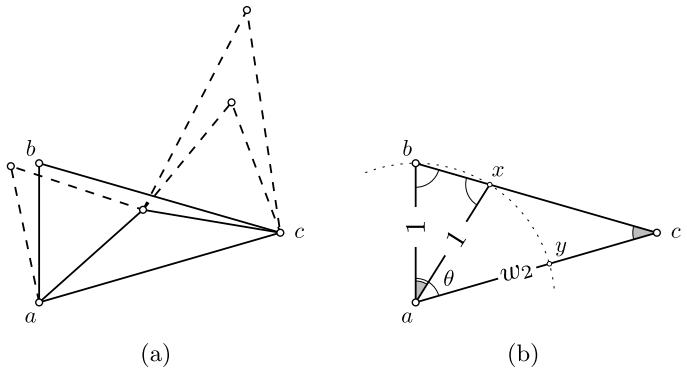

Fig. 18. Two illustrations for the proof of Lemma 10, for the case in which $\triangle_2$ is a tall isosceles triangle and $\triangle_3$ is a flat isosceles triangle. (a) The case in which $v = c$ and $\triangle_4$ (in dashed lines) is flat isosceles or tall isosceles, and the case in which $v = a$ and $\triangle_4$ (in dashed lines) is small equilateral. (b) A focus on the case in which $v = a$ and $\triangle_4$ is small equilateral.

5.2. Conflicts between leaf triangles

Hereafter, we assume that the decomposition tree T of G is rooted at any 3-cycle whose area is maximum. Hence, the triangle realizing such a 3-cycle cannot be drawn inside any other triangle in a planar straight-line realization of G.

The *framework* of G is the subgraph $G_F \subseteq G$ obtained as follows: For each leaf triangle $\triangle_i$ that can be drawn inside a triangle $\triangle_j$, which is either its parent or one of its siblings, we remove from G the vertex v that $\triangle_i$ does not share with $\triangle_j$, along with the two edges incident to v. For example, in Fig. 19, the triangles $\triangle_2$ and $\triangle_4$ can be drawn inside their parent triangle $\triangle_1$, and the triangle $\triangle_{21}$ can be drawn inside its sibling triangle $\triangle_{22}$. Note that G_F is a 2-tree. Further, by Lemma 10, in any planar straight-line realization Γ of G, the triangles we removed from G in order to define G_F are the only triangles that can be drawn inside other triangles representing 3-cycles of G. It follows that the restriction of Γ to G_F is an outerplanar drawing. Hence, we start by testing if G_F is outerplanar, which can be done in linear time [25,42,54]; in the negative case, we reject the instance. In the positive case, we test whether G_F admits a planar straight-line realization respecting its unique outerplane embedding $\mathcal{E}$. Both the computation of $\mathcal{E}$ and the test can be performed in linear time using the algorithms in [25,42,54] and Theorem 4, respectively. If the test is negative we reject the instance, otherwise we have a planar straight-line realization Γ_F of G_F with embedding $\mathcal{E}$.

Let $L_\triangle$ denote the set of triangles that were removed from G to obtain G_F. The third central ingredient of our algorithm is a characterization of the conditions under which Γ_F can be extended to a planar straight-line realization of G, by drawing in Γ_F all the triangles of $L_\triangle$. We start with the following observation.

Observation 12. *Every triangle in $L_\triangle$ is either a small equilateral triangle or a flat isosceles triangle.*

Proof. The observation follows from statement (b) of Lemma 9. □

Let $\triangle = (a, b, c)$ be a triangle of $L_\triangle$ such that $e = (a, b)$ is the unique edge shared by $\triangle$ and G_F. Suppose we want to add a drawing of $\triangle$ to Γ_F. If e is incident to two internal faces of Γ_F, then we call each of the two possible drawings of $\triangle$ an *internal embedding*, regardless of whether a, b, c appear in clockwise or counter-clockwise order when traversing $\triangle$. Consider that e is instead incident to one internal face and to the outer face of Γ_F. Suppose that a precedes b in a counter-clockwise traversal of the outer face of Γ_F. If $\triangle$ is drawn such that a, b, c appear in clockwise order when traversing $\triangle$, then we call such drawing of $\triangle$ an *outer embedding*. Otherwise, we call such drawing of $\triangle$ an *internal embedding*.

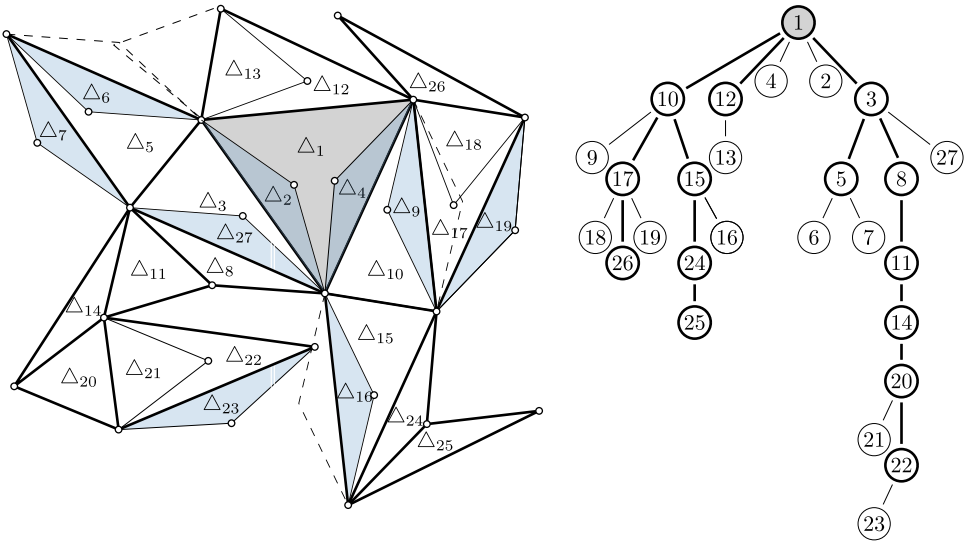

Fig. 19. A planar straight-line realization of a weighted 2-tree for $\sqrt{3} < w_2 \leq 2\cos(15°)$. On the left, the planar straight-line realization. Triangles in the framework are drawn with thick lines. On the right, the decomposition tree rooted at $\triangle_1$.

Consider a straight-line realization Γ of G that *extends* Γ_F; that is, whose restriction to G_F is Γ_F. Suppose that Γ is not planar. We now classify the possible crossings involving triangles in $L_\triangle$.

First, suppose there is a triangle $\triangle \in L_\triangle$ whose drawing in Γ crosses an edge of Γ_F. If the drawing of $\triangle$ is an outer embedding, then we say it induces a *framework conflict*. If the drawing of $\triangle$ is instead an internal embedding, then we say it induces an *overlapping conflict*. Second, consider two triangles $\triangle_i$ and $\triangle_j$ of $L_\triangle$. If the drawings of $\triangle_i$ and $\triangle_j$ in Γ are both outer embeddings, neither of them induces a framework conflict, and they cross each other, then we say that such drawings induce an *external conflict*. Similarly, if the drawings of $\triangle_i$ and $\triangle_j$ in Γ are both internal embeddings, neither of them induces an overlapping conflict, and they cross each other, then we say that such drawings induce an *internal conflict*. In both cases, we say that the drawings of $\triangle_i$ and $\triangle_j$ *conflict with each other*.

We now provide a series of properties of the set of leaf triangles of G. We start with the following lemma, in which we characterize the existence of internal conflicts.

Lemma 11. *Let $\triangle_i$ and $\triangle_j$ be two triangles in $L_\triangle$, and Γ be a straight-line realization of G that extends Γ_F. Suppose that there exists a triangle $\triangle \in \Gamma_F$ such that both $\triangle_i$ and $\triangle_j$ are drawn inside $\triangle$ in Γ. The drawings of $\triangle_i$ and $\triangle_j$ induce an internal conflict if at least one of the following statements is true:*

(a) $\triangle_i$ and $\triangle_j$ share an edge.
(b) $\sqrt{3} < w_2 \leq 2\cos(15°) \simeq 1.93$ and $\triangle$ is a tall isosceles triangle.
(c) $1 < w_2 \leq \sqrt{3}$.

Proof. By statement (a) of Lemma 9 and since each of $\triangle_i$ and $\triangle_j$ is drawn inside $\triangle$, the triangle $\triangle$ is either tall isosceles or big equilateral. Furthermore, by Observation 12, each of $\triangle_i$ and $\triangle_j$ is either small equilateral or flat isosceles. By Lemma 10, the triangles $\triangle_i$ and $\triangle$ share an edge and so do $\triangle_j$ and $\triangle$.

Consider first the case in which $\triangle_i$ shares an edge with $\triangle_j$, as described in statement (a). Note that, since G is a 2-tree, the triangles $\triangle_i$, $\triangle_j$, and $\triangle$ actually share the same edge. By statement (a) of Lemma 9, $\triangle_i$ and $\triangle_j$ cannot be drawn inside each other regardless of their types. Since $\triangle_i$ and

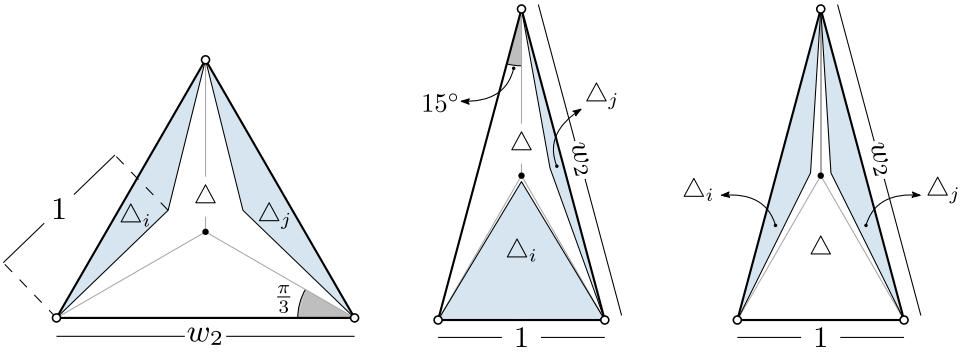

Fig. 20. Illustration for the proof of Lemma 11. On the left, the case in which $\triangle$ is big equilateral. On the right, the case in which $\triangle$ is tall isosceles.

$\triangle_j$ are drawn inside the same triangle and they cannot be drawn inside each other, their drawings induce an internal conflict inside $\triangle$. This proves statement (a).

Consider now the case in which $\triangle_i$ and $\triangle_j$ do not share an edge. The triangles $\triangle_i$ and $\triangle_j$ induce an internal conflict inside $\triangle$ depending on the types of $\triangle_i$, $\triangle_j$, $\triangle$, and the value of w_2. Note that w_2 is necessarily less than 2 since, otherwise, by Property 9 both $\triangle_i$ and $\triangle_j$ would be small equilateral triangles and they could not share each a different side of $\triangle$, since $\triangle$ has at most one side of length 1. By a simple geometric argument we can deduce that two triangles are drawn inside $\triangle$ without inducing internal conflicts in the following cases (refer to Fig. 20):

(i) If $2\cos(15°) \simeq 1.93 < w_2 < 2$ and $\triangle$ is tall isosceles, then two flat isosceles triangles and one small equilateral triangle can be drawn inside $\triangle$ without inducing internal conflicts.
(ii) If $\sqrt{3} \simeq 1.73 < w_2 < 2$ and $\triangle$ is big equilateral, then three flat isosceles can be drawn inside $\triangle$ without inducing internal conflicts; see also Lemma 1.

The limit values $w_2 = 2\cos(15°)$ and $w_2 = \sqrt{3}$ split the interval $1 < w_2 < 2$ into the following three intervals. First, if $2\cos(15°) < w_2 < 2$, then the cases (i) and (ii) are both possible. Hence, $\triangle_i$ and $\triangle_j$ can be drawn inside $\triangle$ without inducing internal conflicts, regardless of their types. Second, if $\sqrt{3} < w_2 \leq 2\cos(15°)$ then the case (i) is no longer possible. Hence, as described in statement (b), $\triangle_i$ and $\triangle_j$ induce an internal conflict inside $\triangle$ if $\triangle$ is a tall isosceles triangle, regardless of the types of $\triangle_i$ and $\triangle_j$. Finally, if $1 < w_2 \leq \sqrt{3}$, then none of the cases is possible. Hence, as described in statement (c), $\triangle_i$ and $\triangle_j$ induce an internal conflict inside $\triangle$ regardless of their types. □

From Lemma 11 we obtain the following necessary condition for the existence of a planar straight-line realization of G extending Γ_F.

Lemma 12. *If there exists a planar straight-line realization of G, then no subset of three triangles of $L_\triangle$ share a side.*

Proof. Suppose, for a contradiction, that there exists a planar straight-line realization Γ of G and that three triangles of $L_\triangle$ share an edge e. Consider two of such triangles that are on the same side of e in Γ and denote them by $\triangle_i$ and $\triangle_j$. If $\triangle_i$ and $\triangle_j$ do not cross each other in Γ, then they are contained one inside the other. However, this is not possible: by Observation 12, $\triangle_i$ and $\triangle_j$ are either small equilateral or flat isosceles triangles, and by statement (a) of Lemma 9, we have that $\triangle_i$ and $\triangle_j$ cannot be drawn inside one another. See Fig. 19. □

If $L_\triangle$ satisfies the condition stated in Lemma 12, we say that $L_\triangle$ is *consistent*.

Lemma 13. *If $L_\triangle$ is consistent, then there are $O(1)$ pairs of triangles of $L_\triangle$ whose drawings induce an internal conflict inside a triangle $\triangle \in \Gamma_F$.*

Algorithm 1: Overview of the $O(n)$-time algorithm described in the proof of Theorem 11.

Input: An n-vertex 2-tree G, in which each edge is assigned one out of two real weights $w_1 = 1$ and $w_2 > 1$.

Result: `True` and a planar straight-line realization of G, or `False` if such a realization does not exist.

STAGE ONE: COMPUTING AND DRAWING THE FRAMEWORK.

Step A: Compute the framework G_F, and the set $L_\triangle$ of leaf triangles of G.
Step B: **if** $L_\triangle$ *is consistent* **then** continue; **else return** `False`;
Step C: **if** G_F *is outerplanar* **then** continue; **else return** `False`;
Construct the unique outerplane embedding $\mathcal{E}$ of G_F.
if G_F *admits a planar straight-line realization with* $\mathcal{E}$ **then** continue; **else return** `False`;
Construct the planar straight-line realization Γ_F of G_F.

STAGE TWO: DRAWING THE SET OF LEAF TRIANGLES ON THE FRAMEWORK.

if $w_2 \geq 2$ **then** // `there are no flat-isosceles triangles`
  Step D.1: Draw the leaf triangles of $L_\triangle$ on Γ_F.
    if *The resulting drawing is planar* **then**
      return `True` *and the constructed realization*;
    else
      return `False`;
    end
else
  Step D.2: Compute all the pairs of triangles of $L_\triangle$ that induce an internal conflict.
  Step E : Compute all the triangles of $L_\triangle$ that induce an overlapping conflict.
  Step F : Compute all the pairs of triangles of $L_\triangle$ that induce an external conflict, and all the triangles of $L_\triangle$ that induce a framework conflict.
  Step G : Test whether Γ_F can be extended to a planar straight-line realization Γ of G.
    if *The realization* Γ *exists* **then**
      Construct Γ and **return** `True` *and* Γ;
    else
      return `False`
    end
end

Proof. By Lemma 10, all the triangles of $L_\triangle$ that can be drawn inside $\triangle$ (including those inducing an internal conflict) share an edge with $\triangle$. By Lemma 12 and the assumed consistency of $L_\triangle$, there are at most 6 of such triangles. Hence, the drawings of at most $\binom{6}{2} = 15$ pairs of triangles can induce an internal conflict inside $\triangle$. □

5.3. The algorithm

We are now ready to describe the algorithm to solve the FEPR problem in $O(n)$ time when the edges of G can only have one of two lengths. The algorithm consists of two stages. In the first stage, we compute the framework G_F of G, and test whether G_F admits a planar straight-line realization Γ_F. Such a realization is constructed in the positive case, and the input instance is rejected otherwise. In the second stage, we test whether Γ_F can be extended to a planar straight-line realization Γ of G, by drawing the triangles of $L_\triangle$ without inducing conflicts. In the positive case, we construct and report Γ as the desired realization of G. As before, the input instance is rejected in the negative case. An overview of the algorithm is shown in Algorithm 1. We describe the algorithm in detail next.

5.3.1. Computing a planar straight-line realization Γ_F of the framework of G.
The first stage of the algorithm is formed by the following three steps.

Step A. Computing the framework G_F of G and the set $L_\triangle$ of leaf triangles. To compute the framework of G we first choose, among all the triangles of G, a triangle $\triangle_O$ with maximum area. Note that $\triangle_O$ cannot be drawn inside any other triangle of G, hence it does not belong to $L_\triangle$. We then compute the decomposition tree T of G rooted at $\triangle_O$. Note that, if G admits a planar straight-line realization, then the degree of T is bounded by a constant. Indeed, each edge of G can be incident to at most two interesting triangles of the same type (and hence each 3-cycle of G can be adjacent to at most 24 distinct 3-cycles). We test whether the degree of T is bounded by 24 in $O(n)$ time. We reject the instance in the negative case. In the positive case we traverse T looking for leaf triangles that can be drawn inside their parent or sibling triangles. By Lemma 10, these are the only triangles in G that can be drawn inside any other triangle. We identify such triangles by means of Observation 3. Since the degree of T is in $O(1)$, each triangle has a constant number of sibling triangles, hence we can compute both G_F and $L_\triangle$ in $O(n)$ time.

Step B. Testing if $L_\triangle$ is consistent. The next step is to test if $L_\triangle$ is consistent. By Lemma 12, this is a necessary condition for the existence of a planar straight-line realization of G. This condition can be trivially tested in $O(n)$ time by looking at the triangles of $L_\triangle$ sharing an edge with a parent or a sibling triangle in T. The input instance is rejected in the negative case.

Step C. Constructing a planar straight-line realization Γ_F of G_F. The final step of this stage is to compute a planar straight-line realization of G_F, if it exists. We first test whether G_F is outerplanar. If this is not the case we reject the instance. Otherwise, we construct its unique outerplane embedding in $O(n)$ time [25,42,54]. Then, by means of Theorem 7 we test in $O(n)$ time if G_F admits a planar straight-line realization with the computed embedding. We construct Γ_F in the positive case, and reject the input instance otherwise.

5.3.2. Extending Γ_F to a planar straight-line realization of G
In the second stage the algorithm distinguishes two cases, namely, the one in which $w_2 \geq 2$ and the one in which $w_2 < 2$. In the former case, which is much simpler to handle, the algorithm is completed by means of a single step, called D.1, while in the latter case the algorithm is completed by means of four steps, called D.2, E, F, and G.

Case $\mathbf{w_2 \geq 2}$. Note that, by Property 9, in this case there are no flat isosceles triangles. Hence, every leaf triangle $\triangle$ in $L_\triangle$ is a small equilateral triangle such that the parent or a sibling of $\triangle$ is a tall isosceles triangle. We act as described in the following step.

Step D.1. Consider each edge e of G_F incident to a triangle in $L_\triangle$. If e is incident to two triangles in $L_\triangle$, we draw them in Γ_F on opposite sides of e. If e is incident to one triangle $\triangle \in L_\triangle$, then we draw $\triangle$ inside any tall isosceles triangle incident to e (recall that such a tall isosceles triangle is the parent or a sibling of $\triangle$). At the end of this process we obtain a straight-line realization Γ of G, and a plane embedding $\mathcal{E}$ of G.

Lemma 14. *G admits a planar straight-line realization if and only if Γ is planar.*

Proof. Obviously, if Γ is planar, then it is a planar straight-line realization of G.

We prove that, if G admits a planar straight-line realization, then Γ is planar. Consider an edge $e \in G_F$. Since $L_\triangle$ is consistent, the edge e is shared by a set $L^e_\triangle \subseteq L_\triangle$ of at most two leaf triangles. Suppose that e is an internal edge of Γ_F. Then, in Γ, each small equilateral triangle $L^e_\triangle$ is drawn as an internal embedding inside a tall isosceles triangle $\triangle^* \in G_F$. Note that no other triangle is drawn inside $\triangle^*$, since a tall isosceles triangle has a single side of length 1. Hence, no triangle of $L^e_\triangle$ creates a crossing in Γ.

Suppose now that e is incident to the outer face of Γ_F. If $L^e_\triangle$ consists of a single triangle, then the triangle is drawn as an internal embedding and, as above, it does not create crossings in Γ. If $L^e_\triangle$ consists instead of two triangles, then least one of them, say $\triangle_s$, is drawn as an outer embedding. However, the two triangles in $L^e_\triangle$ are drawn on different sides of e in any planar straight-line

realization of G, hence if $\triangle_s$ creates a crossing, it does so in every straight-line realization of G. Since a planar straight-line realization of G exists by hypothesis, it follows that $\triangle_s$ does not create crossings in Γ, and hence Γ is planar. □

Observe that Γ and $\mathcal{E}$ are constructed in linear time. We test whether Γ is a planar straight-line realization of G with embedding $\mathcal{E}$ in $O(n)$ time by means of Theorem 4. By Lemma 14, in the negative case we conclude that G admits no planar straight-line realization, otherwise Γ is the desired planar straight-line realization of G.

Case $w_2 < 2$. The algorithm in this case is much more complicated, since there might be flat isosceles triangles in G, which might or might not need to be embedded inside their adjacent large equilateral or tall isosceles triangles in Γ_F. Moreover, more than one leaf triangle might be drawn inside the same interesting triangle of Γ_F. Further, while two leaf triangles might be individually drawn inside the same interesting triangle of Γ_F, it might be the case that they cannot be drawn inside the interesting triangle simultaneously, hence the algorithm might need to perform a choice. This situation differs from the one of the case $w_2 \geq 2$, as in that case a single leaf triangle can be drawn inside an interesting triangle of Γ_F.

An example of a planar straight-line realization of a weighted 2-tree for which $w_2 < 2$ is shown in Fig. 19. The value of w_2 was chosen so that $\sqrt{3} \simeq 1.73 < w_2 \leq 2\cos(15°) \simeq 1.93$. By Property (b) of Lemma 11, a pair of flat isosceles triangles can be drawn inside a big equilateral triangle without crossing each other, but any pair of interesting triangles (either two flat isosceles or a flat isosceles and a small equilateral) cross with each other if they are drawn inside a tall isosceles triangle. The triangles of Γ_F are drawn with thick lines, and the edges of the leaf triangles of $L_\triangle$ that are not part of Γ_F are drawn with thin lines. Observe that any type of interesting triangle can belong to the framework, whereas the triangles of $L_\triangle$ are either small equilateral or flat isosceles triangles. If a leaf triangle is small equilateral, then it is adjacent to (at least) an interesting triangle of the frame that is tall isosceles. If it is instead a flat isosceles, then it is adjacent to (at least) an interesting triangle of the frame that is either tall isosceles or big equilateral.

From Lemma 10, a leaf triangle can be drawn inside its parent triangle, a sibling triangle, or both. In our example, $\triangle_2$ can be drawn both inside its parent triangle $\triangle_1$ and inside the sibling triangle $\triangle_3$. On the other hand, $\triangle_{18}$ can only be drawn inside its parent triangle $\triangle_{17}$, and $\triangle_{21}$ can only be drawn inside its sibling triangle $\triangle_{22}$. Consider the triangle $\triangle_{27}$. It can only be drawn inside its parent triangle $\triangle_3$ since it is congruent to its sibling triangle $\triangle_8$. Hence it would induce an overlapping conflict if it were drawn on the same side as $\triangle_8$. Consider now the interesting triangles drawn with dashed lines. If $\triangle_9$ was drawn inside $\triangle_{17}$, then $\triangle_9$ and $\triangle_{18}$ would cross each other. On the other hand, if the drawing of $\triangle_{16}$ is an outer embedding, then it would induce a framework conflict. Finally, if the drawings of both $\triangle_{13}$ and $\triangle_6$ are outer embeddings, then they would induce an external conflict.

To deal with this case we proceed as follows. First, we compute the pairs of triangles whose drawings may induce a conflict. Achieving this in total linear time is a challenging task, since there might be a linear number of triangles and hence a quadratic number of pairs of triangles. However, we can prove that it suffices to consider only a linear number of pairs of triangles in order to detect all conflicts. Then we construct a 2SAT formula ϕ that contains a Boolean variable for each triangle $\triangle \in L_\triangle$. The value of the variable is associated to a choice of the two possible drawings of $\triangle$: recall that an edge of $\triangle$ is part of G_F, while the vertex of $\triangle$ not incident to such an edge might be placed on either side of it. For each triangle $\triangle$ that induces a framework or an overlapping conflict in one of its two drawings, there is a clause in ϕ that is `True` if and only if the drawing of the triangle is not the one inducing a conflict. Similarly, for each pair of triangles whose drawings induce an internal or an external conflict, there is a clause in ϕ that is `True` if and only if the drawings of the triangles are not both the ones inducing a conflict. If ϕ is satisfiable, then there is a planar straight-line realization of G which can be obtained by embedding the triangles of $L_\triangle$ according to the values of the variables of ϕ. We reject the instance otherwise. We describe the algorithm in detail next.

Step D.2. Detecting internal conflicts. We start by computing all the pairs of triangles of $L_\triangle$ that induce an internal conflict. This is done by traversing T while processing each triangle $\triangle \in \Gamma_F$ as follows. We first obtain the set $L'_\triangle \subseteq L_\triangle$ of leaf triangles that can be drawn inside $\triangle$. By Lemma 10, these triangles share an edge with $\triangle$, so they are children or siblings of $\triangle$. We then decide which pairs of leaf triangles of $L'_\triangle$ induce an internal conflict inside $\triangle$, by means of Lemma 11. Since $L_\triangle$ is consistent, by Lemma 13 there are $O(1)$ triangles in $L'_\triangle$. Hence, we process each triangle in $O(1)$ time, and the traversal takes $O(n)$ time.

Step E Detecting overlapping conflicts. Next, for each triangle $\triangle \in L_\triangle$, we detect whether any of its two drawings induces an overlapping conflict. Let $\triangle^*$ be the parent of $\triangle$ in T and let e be the edge shared by $\triangle$ and $\triangle^*$. Note that, since $\triangle \in L_\triangle$, one of the two drawings of $\triangle$ lies inside $\triangle^*$ and hence does not induce an overlapping conflict. We first check whether the other drawing of $\triangle$ is an internal or an outer embedding (in the latter case, we can conclude that $\triangle$ does not induce an overlapping conflict). This is equivalent to checking whether there is a triangle $\triangle_s$ in Γ_F on the other side of e with respect to $\triangle^*$. Note that, if such a triangle $\triangle_s$ exists, then it is a sibling of $\triangle$ in T, given that it shares the edge e with $\triangle$ and that it is not the parent $\triangle^*$ of $\triangle$. Hence, for each of the $O(1)$ siblings of $\triangle$ in T, we check in $O(1)$ time whether it is a triangle in Γ_F and whether it is incident to e. If no sibling of $\triangle$ in T satisfies both checks, then we conclude that no drawing of $\triangle$ induces an overlapping conflict. Otherwise, there is a sibling of $\triangle$ in T that satisfies both checks; then this sibling is the desired triangle $\triangle_s$ and hence the drawing of $\triangle$ outside $\triangle^*$ is an internal embedding of $\triangle$. Then we also check in $O(1)$ time whether $\triangle_s$ and the drawing of $\triangle$ outside $\triangle^*$ cross each other. In the positive case, we conclude that the drawing of $\triangle$ outside $\triangle^*$ induces an overlapping conflict, while in the negative case we conclude that it does not. Since we process each triangle $\triangle \in L_\triangle$ in $O(1)$ time, this step takes $O(n)$ time.

Step F. Detecting framework and external conflicts. Detecting framework and external conflicts in linear time is more challenging than overlapping and internal conflicts. Indeed, an outer embedding of a triangle $\triangle \in L_\triangle$ might cross a triangle of Γ_F or the outer embedding of a triangle in $L_\triangle$ that is far, in terms of graph-theoretic distance, from $\triangle$. Hence, the trivial solution would detect framework and external conflicts by comparing each triangle $\triangle \in L_\triangle$ with every triangle of Γ_F and with every triangle in $L_\triangle$. However, this would result in a quadratic running time. In order to achieve linear running time, we decompose the plane into cells and then derive several combinatorial properties of such a decomposition, including: (i) only a linear number of cells can host conflicts and such cells can be found efficiently; (ii) conflicts can only occur between triangles in a single cell or in adjacent cells; and (iii) each cell can host only a constant number of outer embeddings of triangles from $L_\triangle$. The last property is only true because of the assumption $w_2 < 2$. We now describe the details of Step F.

Let $L'_\triangle$ be the subset of $L_\triangle$ composed of those triangles that are incident to external edges of Γ_F. Only the outer embeddings of the triangles in $L'_\triangle$ can induce framework and external conflicts. We translate the Cartesian axes so that the bottom-left corner of the bounding box of Γ_F lies on the origin, and give an outer embedding to every triangle in $L'_\triangle$. This results in a possibly non-planar straight-line realization Γ'_F of the graph $G'_F := G_F \cup L'_\triangle$. Our strategy to detect external and framework conflicts is based on the construction of a linear-size "proximity" graph H defined on the vertices of G'_F. We show that, if two triangles of $L'_\triangle$ induce an external conflict, then their vertices are associated either with the same node or with adjacent nodes of H. Similarly, if a triangle $\triangle \in L'_\triangle$ induces a framework conflict by intersecting an edge e of G'_F, then the vertices of $\triangle$ and the end-vertices of e are associated either with the same node or with adjacent nodes of H. We then detect external and framework conflicts by a linear-time traversal of H.

In the following, we say that a vertex of G_F is a *framework vertex* and a vertex of $L'_\triangle$ that is not contained in G_F is a *leaf vertex*. For each leaf vertex v, we denote with $\triangle(v)$ the outer embedding of the triangle in $L'_\triangle$ whose leaf vertex is v. We define the graph H as follows. Consider a square grid whose cells have side length $3w_2$ covering the plane; see Fig. 21. We assign a label $l(v) = (i, j)$ to every vertex $v \in V(G'_F)$, where i and j respectively denote the row and the column of one of the (at most four) grid cells that contain v in their interior or on their boundary. More precisely, the vertex v gets the label $(\lfloor \frac{x(v)}{3w_2} \rfloor, \lfloor \frac{y(v)}{3w_2} \rfloor)$, where $(x(v), y(v))$ denotes the coordinates of v in Γ'_F. The graph H

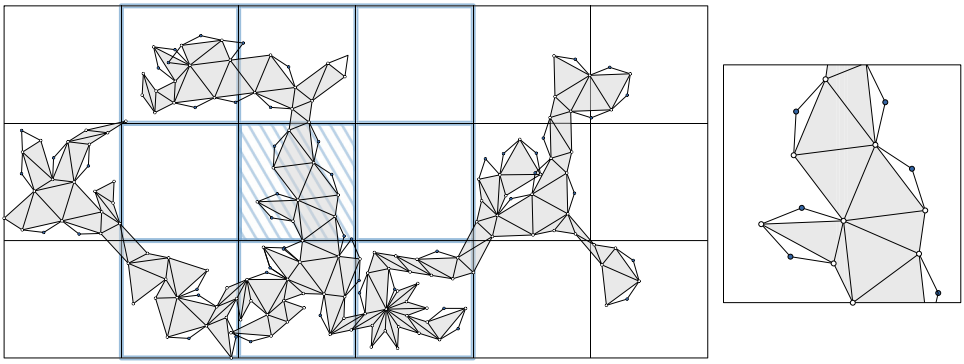

Fig. 21. On the left, the straight-line realization Γ'_F of the graph $G'_F := G_F \cup L'_\triangle$, where each triangle in $L'_\triangle$ has an outer embedding. The triangles of G_F are shown in gray, and the outer embeddings of the triangles in $L'_\triangle$ are shown in white. Framework vertices are shown in white and leaf vertices are shown in blue. The grid cell filled with dashed lines is zoomed-in on the right. All the triangles in $L'_\triangle$ contained in this cell can induce external or framework conflicts only with the triangles contained in the nine highlighted cells.

has a node for each label assigned to at least one vertex of G'_F, and two nodes (i, j) and (i', j') are connected if and only if $|i - i'| \leq 1$ and $|j - j'| \leq 1$. Note that H has at most n nodes and maximum degree 8, hence it has $O(n)$ edges. We say that a vertex of G'_F is *associated with* a node μ of H if μ represents the cell (i, j) and $l(v) = (i, j)$.

Lemma 15. *The graph H satisfies the following properties:*

(a) For each node μ of H, there are $O(1)$ leaf vertices associated with μ.
(b) Let $\triangle_1, \triangle_2$ be two triangles in $L'_\triangle$. For $i = 1, 2$, let μ_i denote a node of H such that at least one vertex of $\triangle_i$ is associated with μ. If the outer embeddings of $\triangle_1$ and $\triangle_2$ induce an external conflict, then either $\mu_1 = \mu_2$ or $(\mu_1, \mu_2) \in E(H)$.
(c) Let $\triangle$ be a triangle in $L'_\triangle$, and e be an edge of G_F incident to the outer face of Γ_F. Let μ and ν denote two nodes of H such that at least one vertex of $\triangle$ is associated with μ and at least one end-vertex of e is associated with ν. If the outer embedding of $\triangle$ induces a framework conflict by intersecting e, then either $\mu_1 = \mu_2$ or $(\mu_1, \mu_2) \in E(H)$.

Proof. Consider any leaf vertex v associated with a node μ of H. Let (i, j) be the cell corresponding to μ and $\triangle$ be the parent triangle of the triangle in $L'_\triangle$ that has v as a vertex. By statement (b) of Lemma 9, we have that $\triangle$ is either a big equilateral or a tall isosceles triangle. Further, $\triangle$ is contained in the union of the nine cells (i', j') for $|i - i'| \leq 1$ and $|j - j'| \leq 1$ (see the highlighted cells in Fig. 21); this is because the Euclidean distance between v and every point of $\triangle$ is smaller than $2w_2$, while the side length of a grid cell is $3w_2$. Observe that the area of a big equilateral or tall isosceles triangle is larger than the area of a small equilateral triangle, which is equal to $\sqrt{3}/4$. Moreover, any two parent triangles are interior-disjoint in Γ'_F. Since the area of the union of the nine cells is $81w_2^2$, then at most $81w_2^2 \frac{4}{\sqrt{3}} = \frac{324}{\sqrt{3}} w_2^2 \in O(w_2^2)$ parent triangles are contained in the nine cells, and $O(w_2^2)$ leaf vertices are associated with μ. Since $w_2 < 2$, we have $O(w_2^2) \in O(1)$. This proves Property (a).

To prove Property (b), suppose, for the sake of contradiction, that the outer embeddings of $\triangle_1$ and $\triangle_2$ induce an external conflict, with $\mu_1 \neq \mu_2$ and $(\mu_1, \mu_2) \notin E(H)$. Since $\triangle_1$ and $\triangle_2$ induce an external conflict, then there is at least a point x in $\triangle_1 \cap \triangle_2$. Note that x is at distance smaller than or equal to w_2 from the vertices of $\triangle_1$ and $\triangle_2$, which implies that every vertex of $\triangle_1$ is at distance smaller than or equal to $2w_2$ from any vertex of $\triangle_2$. Since there is a distance larger than or equal to $3w_2$ between two vertices of G'_F associated to non-adjacent nodes of H, then either $\mu_1 = \mu_2$ or $(\mu_2, \mu_2) \in E(H)$, which is a contradiction.

Finally, Property (c) can be proved by similar arguments to those we used to prove Property (b). □

In order to detect external and framework conflicts, we define each node μ of H representing the cell (i, j) as a record storing (i) a label $l(\mu) = (i, j)$ and (ii) a list $L(\mu)$ that contains the vertices of G'_F associated with μ. We next describe an efficient algorithm to construct H.

Lemma 16. *The graph H can be constructed in $O(n)$ time.*

Proof. We first show how to construct the vertex set $V(H)$ of H.

Consider a total order π of the vertices of G'_F such that, for any two vertices u and v with $l(u) = (i_u, j_u)$ and $l(v) = (i_v, j_v)$, it holds that u precedes v in the order if (i) $i_u < i_v$ or (ii) $i_u = i_v$ and $j_u < j_v$; if $i_u = i_v$ and $j_u = j_v$, then u and v are in any relative order in π. Since G'_F is connected and any edge of G'_F has length at most w_2, then $i, j \leq \frac{w_2 n}{3w_2} = \frac{1}{3}n$ for any label (i, j). Hence, we can compute π in $O(n)$ time by means of counting sort.

We process the vertices of G'_F in the order in which they appear in π. Let s be the first vertex in the order π. We create a node ν of H, set $l(\nu) = l(s)$, and add s to $L(\nu)$. Let now v be the currently considered vertex of π and let us denote by μ the last created node of H. If $l(v) = l(\mu)$, then we add v to $L(\mu)$; otherwise, we create a node μ' of H, set $l(\mu') = l(v)$, and add v to $L(\mu')$. Note that, if two vertices of G'_F have the same label, then they are consecutive in π. Therefore, the above procedure creates a new node of H for each label of a vertex of G'_F, and vertices with the same label (i, j) are all placed in the list $L(\mu)$ of the node μ such that $l(\mu) = (i, j)$. The procedure can clearly be performed in overall $O(n)$ time.

We now show how to construct the edge set $E(H)$ of H.

We define a partition of $E(H)$ into four subsets $E_-, E_|, E_/, E_\backslash$; refer to Fig. 22. Consider any edge $(\mu, \nu) \in E(H)$, where we denote $l(\mu) = (i_\mu, j_\mu)$ and $l(\nu) = (i_\nu, j_\nu)$. Then (μ, ν) belongs to E_-, $E_|$, $E_/$, or $E_\backslash$, depending on whether (μ, ν) satisfies Property P_-, $P_|$, $P_/$, or $P_\backslash$, respectively, where:

P_-: $i_\mu = i_\nu$ and $|j_\mu - j_\nu| = 1$,

$P_|$: $j_\mu = j_\nu$ and $|i_\mu - i_\nu| = 1$,

$P_/$: either $i_\mu - i_\nu = j_\mu - j_\nu = 1$ or $i_\mu - i_\nu = j_\mu - j_\nu = -1$, and

$P_\backslash$: either $i_\mu - i_\nu = 1$ and $j_\mu - j_\nu = -1$ or $i_\mu - i_\nu = -1$ and $j_\mu - j_\nu = 1$.

In order to construct the sets $E_-, E_|, E_/$, and $E_\backslash$, we define four total orders $\pi_-, \pi_|, \pi_/$, and $\pi_\backslash$ of the nodes of H, respectively. For any two nodes μ and ν of H, it holds that μ precedes ν in π_-, $\pi_|$, $\pi_/$, and $\pi_\backslash$ if and only if the following conditions hold, respectively:

C_-: either $i_\mu < i_\nu$ or $i_\mu = i_\nu$ and $j_\mu < j_\nu$,

$C_|$: either $j_\mu < j_\nu$ or $j_\mu = j_\nu$ and $i_\mu < i_\nu$,

$C_/$: either $i_\mu - j_\mu < i_\nu - j_\nu$ or $i_\mu - j_\mu = i_\nu - j_\nu$ and $i_\mu < i_\nu$, or

$C_\backslash$: either $i_\mu + j_\mu < i_\nu + j_\nu$ or $i_\mu + j_\mu = i_\nu + j_\nu$ and $i_\mu < i_\nu$.

We construct the set E_i with $i \in \{-, |, /, \backslash\}$ as follows. First, we compute the total order π_i of the nodes of H for which condition C_i holds for any pair (μ, ν) of nodes such that μ precedes ν in π_i. Similarly as in the construction of π, the order π_i can be computed in $O(n)$ time by means of counting sort. Then we process the nodes of $V(H)$ in the order π_i and add a new edge (μ, ν) to E_i whenever condition P_i holds for two nodes μ and ν such that μ immediately precedes ν in π_i. Since, by construction, nodes that are adjacent in E_i are consecutive in π_i, it follows that the above procedure correctly computes each set E_i. Clearly, the procedure can be performed in overall $O(n)$ time. This concludes the construction of H. □

We now describe an algorithm, which we call ConflictFinder, to detect external and framework conflicts. Broadly speaking, the algorithm performs two linear-time traversals of the edges of H. In the first traversal, we detect external conflicts. When processing the edge $(\mu, \nu) \in E(H)$ we

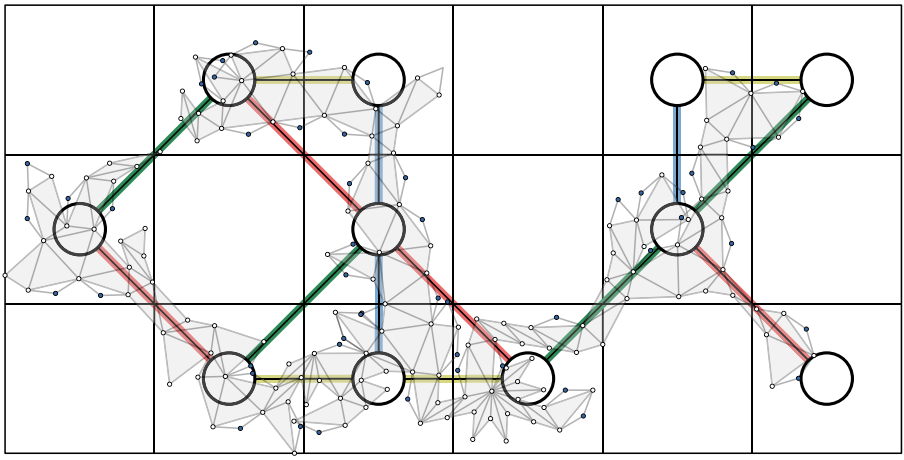

Fig. 22. Construction of the set of edges $E(H)$. The edges on the sets E_{-}, $E_{|}$, $E_{/}$, and $E_{\backslash}$ are shown in yellow, blue, green, and red, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

detect intersections between the triangles of $L'_{\triangle}$ whose leaf vertices are contained in $L(\mu) \cup L(\nu)$. In the second traversal, we detect framework conflicts. In this case, when processing the edge $(\mu, \nu) \in E(H)$ we detect intersections between the triangles of $L'_{\triangle}$ and the edges of G'_F whose leaf and framework end-vertices, respectively, are contained in $L(\mu) \cup L(\nu)$.

The algorithm consists of the following steps:

Step 1: Compute the drawing Γ'_F of G'_F and the auxiliary graph H as described above.

Step 2: For every framework vertex v, create the list $L_E(v)$ of the two edges of Γ_F incident to both v and the outer face of Γ_F.

Step 3: For every node $\mu \in V(H)$, create the lists $L_\ell(\mu)$ and $L_F(\mu)$ of leaf and framework vertices contained in $L(\mu)$, respectively.

Step 4: Detect which triangles of $L'_{\triangle}$ induce external conflicts as follows. Let $\mathcal{E}_\ell = \{(u, v) \in L_\ell(\mu) \times \{L_\ell(\mu) \cup L_\ell(\nu)\} \mid (\mu, \nu) \in E(H)\}$. For every pair of vertices $(u, v) \in \mathcal{E}_\ell$, test if $\triangle(u)$ and $\triangle(v)$ intersect each other. In the positive case, the triangles $\triangle(u)$ and $\triangle(v)$ induce an external conflict.

Step 5: Detect which triangles of $L'_{\triangle}$ induce framework conflicts as follows. Let $\mathcal{E}_F = \{(u, v) \in L_F(\mu) \times \{L_\ell(\mu) \cup L_\ell(\nu)\} \mid (\mu, \nu) \in E(H)\}$. For every pair of vertices $(u, v) \in \mathcal{E}_F$, test if there is an edge $e \in L_E(u)$ such that e intersects $\triangle(v)$. In the positive case, the triangle $\triangle(v)$ induces a framework conflict with the edge e.

The correctness and complexity of the algorithm are proved in the following lemma.

Lemma 17. *The algorithm* ConflictFinder *detects all the external and framework conflicts in $O(n)$ time.*

Proof. We first prove the correctness of the algorithm. Consider first the detection of external conflicts. Let u and v be two leaf vertices associated respectively to two nodes μ and ν. Remember that $\triangle(u)$ and $\triangle(v)$ induce an external conflict if and only if they intersect each other. Then, by Property (b) of Lemma 15, we have that either $\mu = \nu$ or $(\mu, \nu) \in E(H)$. Since in Step 4 the algorithm processes each pair of leaf triangles whose leaf vertices are associated with the same node of H or with adjacent nodes of H, it follows that all the external conflicts are being detected.

Consider now the detection of framework conflicts. Let u be a framework vertex associated to a node μ, let v be a leaf vertex associated to a node ν, and e be an edge in $L_E(u)$. Remember that

$\triangle(v)$ induces a framework conflict with e if and only if $\triangle(v)$ and e intersect each other. Then, by Property (c) of Lemma 15, we have that either $\mu = \nu$ or $(\mu, \nu) \in E(H)$. Since in Step 5 the algorithm processes each pair composed of a framework and a leaf vertex that are associated either with the same node of H or with adjacent nodes of H, it follows that all the framework conflicts are being detected.

We now bound the time complexity of the algorithm. This is done by bounding the time complexity of each of the Steps 1–5.

- Since there are $O(n)$ triangles in $L'_\triangle$, we have that Γ_F can be extended to Γ'_F in $O(n)$ time. Furthermore, the graph H can be computed in $O(n)$ time by Lemma 16. Hence Step 1 takes $O(n)$ time.
- Step 2 takes $O(n)$ time since the boundary of the outer face of Γ_F consists of $O(n)$ line segments.
- Step 3 takes also $O(n)$ time, since H has at most n nodes, and every vertex v of G'_F is contained in the list $L(\mu)$ of a single node μ of H.
- The time complexity of Step 4 is bounded by $O(|\mathcal{E}_\ell|)$. Note that H contains $O(n)$ edges; moreover, for every node μ of H, the list $L_\ell(\mu)$ contains $O(1)$ leaf vertices, by Property (a) of Lemma 15. Hence $\mathcal{E}_\ell$ contains $O(n)$ pairs of vertices and Step 4 takes $O(n)$ time.
- Finally, the time complexity of Step 5 is bounded by $O(|\mathcal{E}_F|)$. In this case, even though $\Theta(n)$ framework vertices might be associated with the same node of H, each framework vertex appears in $O(1)$ pairs of $\mathcal{E}_F$; namely, if a framework vertex u is associated with a node μ of H, then it only appears in the pairs (u, v) of $\mathcal{E}_F$ such that the leaf vertex v is associated either with μ or with one of the (at most 8) neighbors of μ in H. By Property (a) of Lemma 15, there are $O(1)$ leaf vertices associated with the same node of H, hence $\mathcal{E}_F$ contains $O(n)$ pairs of vertices and Step 5 takes $O(n)$ time.

This concludes the proof of the time complexity and hence the proof of the lemma. □

Step G. Extending Γ_F to a planar straight-line realization of G. The final step is to draw the triangles of $L_\triangle$ in Γ_F without inducing conflicts, so that the resulting drawing remains planar. We decide if this is possible by means of a 2SAT formula. Each leaf triangle $\triangle_i \in L_\triangle$ can have two embeddings (which differ from each other for the position of the leaf vertex of $\triangle_i$). We arbitrarily associate each embedding with a distinct truth value of a suitable Boolean variable, and construct the clauses of the formula as follows:

- If a drawing of $\triangle_i$ induces an overlapping conflict, then we introduce a clause $(\neg\ell_i)$, where ℓ_i is the literal that is `True` when $\triangle_i$ has the drawing that induces an overlapping conflict.
- If $\triangle_i$ induces a framework conflict, we introduce a clause $(\neg\ell_i)$, where ℓ_i is the literal that is `True` when $\triangle_i$ has an outer embedding.
- If two leaf triangles $\triangle_i, \triangle_j \in L_\triangle$ induce an internal conflict inside a triangle $\triangle \in \Gamma_F$, we introduce a clause $(\neg\ell_i \vee \neg\ell_j)$, where ℓ_i (resp. ℓ_j) is the literal that is `True` when $\triangle_i$ (resp. $\triangle_j$) is drawn inside $\triangle$.
- If two leaf triangles $\triangle_i, \triangle_j \in L_\triangle$ induce an external conflict, we introduce a clause $(\neg\ell_i \vee \neg\ell_j)$, where ℓ_i (resp. ℓ_j) is the literal that is `True` when $\triangle_i$ (resp. $\triangle_j$) has an outer embedding.

Let ϕ be the 2SAT formula produced by the clauses introduced as described above. Note that the clauses of ϕ describe all and only the overlapping, framework, internal, and external conflicts defined in Section 5.2. Moreover, since there are $O(n)$ triangles in $L_\triangle$, the number of framework and overlapping conflicts is in $O(n)$. Further, since H has $O(n)$ vertices and edges, by Lemmas 13 and 15 the number of internal and external conflicts is in $O(n)$, respectively. It follows that ϕ contains $O(n)$ clauses. The following lemma shows that a solution to ϕ corresponds to a planar straight-line realization of G.

Lemma 18. *The formula ϕ is satisfiable if and only if Γ_F can be extended to a planar straight-line realization of G. Also, if ϕ is satisfiable, then such a planar straight-line realization can be constructed in $O(n)$ time.*

Proof. We prove the two directions separately.
($\Longrightarrow$) Assume that ϕ is satisfiable. We show that Γ_F can be extended to a planar straight-line realization of G. Consider a truth assignment for the Boolean variables that satisfies ϕ. The variables of ϕ are in one to one correspondence with the triangles of $L_\triangle$. We draw in Γ_F each triangle $\triangle_i \in L_\triangle$ with the embedding associated with the truth value of the variable corresponding to $\triangle_i$ in the satisfying assignment for ϕ. This can be done in $O(1)$ time per triangle in $L_\triangle$, hence in overall $O(n)$ time. Since each clause of ϕ is satisfied, it follows that $\triangle_i$ is drawn in Γ_F without inducing conflicts. Since this is true for every $\triangle_i \in L_\triangle$, it follows that the resulting straight-line realization of G is planar. Furthermore, such a realization can be constructed in $O(n)$ time since there are $O(n)$ clauses in ϕ.
($\Longleftarrow$) Let Γ be a planar straight-line realization of G. We show that ϕ is satisfiable. The embedding in Γ of each triangle $\triangle_i \in L_\triangle$ identifies a truth value for the corresponding variable in ϕ, and thus, a truth assignment for ϕ. Such an assignment satisfies ϕ since an unsatisfied clause would imply that we have either an overlapping, or a framework, or an internal, or an external conflict, which in turn implies that the boundary of some triangle $\triangle_i \in L_\triangle$ crosses the boundary of a triangle of Γ_F or the boundary of another triangle of $L_\triangle$, contradicting the fact that Γ is planar. This concludes the proof. □

By the discussion in Steps D.2, E, and F, we have that ϕ can be constructed in $O(n)$ time. Further, as argued above, ϕ contains $O(n)$ clauses. We can therefore test whether ϕ is satisfiable in $O(n)$ time [6]. If ϕ is not satisfiable, we reject the instance. Otherwise, we extend Γ_F to a planar straight-line realization of G in $O(n)$ time as described in the proof of Lemma 18. This concludes the proof of Theorem 11.

6. Maximal outerplanar graphs

In this section we study the FEPR problem for weighted outerplanar 2-trees, that is, for weighted maximal outerplanar graphs. While we could not establish the computational complexity of the FEPR problem for general maximal outerplanar graphs, we can prove the following two theorems.

Theorem 13. *Let G be an n-vertex weighted maximal outerpath. There exists an $O(n)$-time algorithm that tests whether G admits a planar straight-line realization and, in the positive case, constructs such a realization.*

Theorem 14. *Let G be an n-vertex weighted maximal outerpillar. There exists an $O(n^3)$-time algorithm that tests whether G admits a planar straight-line realization and, in the positive case, constructs such a realization.*

Before proving Theorems 13 and 14, we establish some notation and definitions that are common to both proofs. Let G be a weighted 2-tree. For a planar straight-line realization Γ of G, the *interior* of Γ, denoted by $\mathcal{I}_\Gamma$, is the complement of the outer face of Γ, i.e., the union of the closure of the internal faces of Γ. Let e be an edge of G. An *e-outer realization* of G is a planar straight-line realization of G such that e is incident to its outer face. Given two e-outer realizations Γ and Γ' of G, we say that Γ is *e-smaller than Γ'* if there exists an e-outer realization Γ'' that can be obtained by a rigid transformation of Γ such that the end-vertices of e lie at the same points in Γ'' and in Γ' and such that $\mathcal{I}_{\Gamma''} \subseteq \mathcal{I}_{\Gamma'}$ (see Fig. 23); that is Γ is e-smaller than Γ' if it can be rotated, translated, and possibly reflected so that the representation of e coincides with the one in Γ' and so that its interior is a subset of the interior of Γ'. A set $\mathcal{R}$ of e-outer realizations of G is *e-optimal* if, for every e-outer realization Γ' of G, there exists an e-outer realization Γ of G in $\mathcal{R}$ such that Γ is e-smaller than Γ'. An e-outer realization Γ of G is *e-optimal* if the set $\{\Gamma\}$ is e-optimal.

The following lemma will be very useful; see Fig. 24. Roughly speaking, it asserts that, in order to construct a planar straight-line realization of a weighted 2-tree, for certain subgraphs we might assume, without loss of generality, that the realizations are "as e-small as possible".

Lemma 19. *Let G be a weighted 2-tree that admits a planar straight-line realization $\mathcal{G}$. Let H and K be two subgraphs of G such that:*

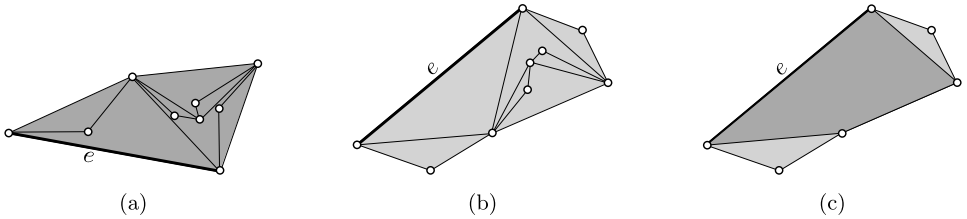

Fig. 23. (a) and (b) show two e-outer realizations Γ and Γ' of a weighted 2-tree. The gray regions show $\mathcal{I}_{\Gamma}$ and $\mathcal{I}_{\Gamma'}$. We have that Γ is e-smaller than Γ'. Indeed, as shown in (c), Γ can be rotated, translated, and reflected so that the representation of e coincides with the one in Γ' and so that its interior is a subset of the interior of Γ'.

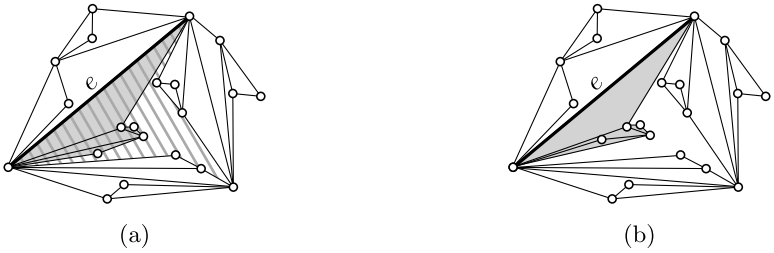

Fig. 24. (a) The drawing $\mathcal{G}$; the set $\mathcal{I}_{\mathcal{H}}$ is shaded gray. The region with dark gray strips is the face f of $\mathcal{K}$. (b) The drawing obtained from $\mathcal{G}$ by replacing $\mathcal{H}$ with $\mathcal{H}''$; the set $\mathcal{I}_{\mathcal{H}''}$ is gray.

(i) $H \cap K = e$, that is, H and K share an edge e, the end-vertices of e, and no other edge or vertex; and
(ii) G is the 2-clique-sum of H obtained by identifying the two copies of e into a single edge e of G.

Suppose also that the restriction $\mathcal{H}$ of $\mathcal{G}$ to H is an e-outer realization of H, and that no vertex of K lies inside an internal face of $\mathcal{H}$ in $\mathcal{G}$.

Let $\mathcal{H}'$ be any e-outer realization of H that is e-smaller than $\mathcal{H}$. Then it is possible to replace $\mathcal{H}$ with an e-outer realization of H that can be obtained by a rigid transformation of $\mathcal{H}'$ in such a way that the resulting straight-line realization of G is planar.

Proof. Let $\mathcal{K}$ be the restriction of $\mathcal{G}$ to K. Assume that H has more than two vertices, as otherwise H contains a single edge, hence $\mathcal{H}'$ can be obtained by a rigid transformation of $\mathcal{H}$ and the statement is trivial. By assumption, no vertex of K lies in $\mathcal{I}_{\mathcal{H}}$; further, H is biconnected, by Property (P1) of a 2-tree. It follows that there is a face f of $\mathcal{K}$ that contains $\mathcal{H}$ entirely, except for the edge e and its end-vertices, which are shared by H and K. By assumption, there exists an e-outer realization $\mathcal{H}''$ of H that can be obtained by a rigid transformation of $\mathcal{H}'$ such that $\mathcal{I}_{\mathcal{H}''} \subseteq \mathcal{I}_{\mathcal{H}}$, hence the part of $\mathcal{H}''$ different from the edge e does not intersect $\mathcal{K}$; the edge e does not intersect $\mathcal{K}$ either, as it is part of it and $\mathcal{K}$ is planar. Hence, the straight-line realization of G resulting from the replacement of $\mathcal{H}$ with $\mathcal{H}''$ in $\mathcal{G}$ is planar. □

We are now ready to prove Theorems 13 and 14.

Proof of Theorem 13. Let $G = (V, E, \lambda)$ be an n-vertex weighted maximal outerpath (as the one in Fig. 25) and let $\mathcal{O}$ be its outerplane embedding, which can be found in $O(n)$ time [25,42,54]. Since G is an outerpath, its dual tree is a path, which we denote by $P := (p_1, \ldots, p_k)$, where $k = n - 2$; note that P can be easily recovered from $\mathcal{O}$ in $O(n)$ time. Each node p_i of P corresponds to an internal face of $\mathcal{O}$; we denote by c_i the 3-cycle of G bounding such a face; each cycle c_i has a unique (planar) straight-line realization (up to a rigid transformation), which we denote by $\mathcal{C}_i$. Further, for $i = 1, \ldots, k-1$, let e_i be the edge of G that is dual to the edge (p_i, p_{i+1}) of P; note that e_i is an internal edge of $\mathcal{O}$.

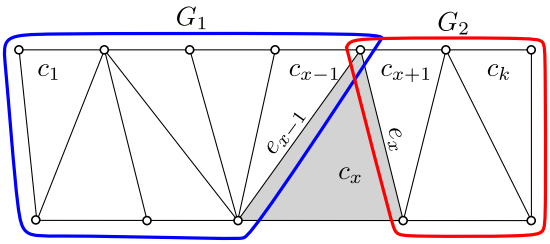

Fig. 25. The outerplane embedding of an outerpath. The gray face is the one bounded by c_x. The graphs G_1 and G_2 are enclosed inside fat curves.

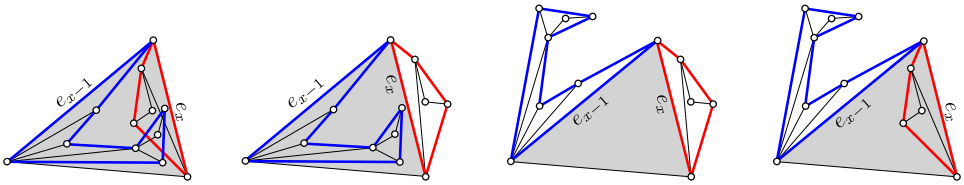

Fig. 26. The four different ways to combine Γ_1 and Γ_2 with $\mathcal{C}_x$. In this example, the first realization is not planar, while the other three are.

Let $x \in \{1, \ldots, k\}$ be an index such that the area of c_x is maximum. Let G_1 be the subgraph of G composed of the 3-cycles $c_1, c_2, \ldots, c_{x-1}$ and let G_2 be the subgraph of G composed of the 3-cycles $c_{x+1}, c_{x+2}, \ldots, c_k$. One of these graphs might be undefined if $x = 1$ or $x = k$; however, in the following, we assume that $1 < x < k$ as the arguments for the cases $x = 1$ and $x = k$ are analogous and actually simpler. Note that G_1 and G_2 are maximal outerpaths.

Assume that a planar straight-line realization Γ of G exists; for $i = 1, 2$, let Γ_i be the restriction of Γ to G_i. Since the area of c_x is maximum among all the 3-cycles of G, it follows that c_x does not lie inside any 3-cycle of G in Γ; hence, Γ_1 is an e_{x-1}-outer realization of G_1 and Γ_2 is an e_x-outer realization of G_2.

A key ingredient of our algorithm is the fact that, if there exists an e_{x-1}-outer realization of G_1 (resp. an e_x-outer realization of G_2), then there exists an e_{x-1}-*optimal* realization of G_1 (resp. an e_x-*optimal* realization of G_2). Indeed, in the following, we show an $O(n)$-time algorithm that either (i) concludes that G_1 admits no e_{x-1}-outer realization or that G_2 admits no e_x-outer realization, and thus G admits no planar straight-line realization, or (ii) constructs an e_{x-1}-optimal realization Γ_1 of G_1 and its plane embedding $\mathcal{E}(\Gamma_1)$, and an e_x-optimal realization Γ_2 of G_2 and its plane embedding $\mathcal{E}(\Gamma_2)$.

In view of the argument above, the existence of such an algorithm suffices to conclude the proof of the theorem. Indeed, if the algorithm concludes that G_1 admits no e_{x-1}-outer realization or that G_2 admits no e_x-outer realization, then G admits no planar straight-line realization. Otherwise, the algorithm constructs an e_{x-1}-optimal realization Γ_1 of G_1, together with its plane embedding $\mathcal{E}(\Gamma_1)$, and an e_x-optimal realization Γ_2 of G_2, together with its plane embedding $\mathcal{E}(\Gamma_2)$. Then, in order to test whether G admits a planar straight-line realization, we consider the unique (up to a rigid transformation) planar straight-line realization $\mathcal{C}_x$ of c_x and we try to combine it with Γ_1 and Γ_2 (see Fig. 26). This combination can be done in four ways. Indeed, once Γ_1 and $\mathcal{C}_x$ coincide on the representation of e_{x-1}, it still remains to decide whether the vertex of c_{x-1} not incident to e_{x-1} and the vertex of c_x not incident to e_{x-1} lie on different sides or on the same side of the line through e_{x-1}, and similarly for Γ_2 and $\mathcal{C}_x$. These two independent choices result in four straight-line realizations of G. Each of these realizations, say Γ, can be constructed in $O(n)$ time; further, its plane embedding $\mathcal{E}(\Gamma)$ can also be constructed in $O(n)$ time from $\mathcal{E}(\Gamma_1)$ and $\mathcal{E}(\Gamma_2)$. Then it can be tested in $O(n)$ time whether Γ is a planar straight-line realization respecting $\mathcal{E}(\Gamma)$, due to Theorem 4. If (at least) one of

the four realizations is planar, then it is the desired planar straight-line realization of G. Otherwise, we claim that there exists no planar straight-line realization of G. For a contradiction, assume that there exists such a realization, and denote it by Γ. Since Γ_1 is e_{x-1}-optimal, it is e_{x-1}-smaller than the restriction of Γ to G_1; hence, we can apply Lemma 19 with $H = G_1$, $K = G_2 \cup c_x$, $\mathcal{G} = \Gamma$, $e = e_{x-1}$, and $\mathcal{H}' = \Gamma_1$, in order to obtain a planar straight-line realization Γ' of G whose restriction to G_1 is Γ_1. Further, since Γ_2 is e_x-optimal, it is e_x-smaller than the restriction of Γ' to G_2; hence, we can apply Lemma 19 with $H = G_2$, $K = G_1 \cup c_x$, $\mathcal{G} = \Gamma'$, $e = e_x$, and $\mathcal{H}' = \Gamma_2$, in order to obtain a planar straight-line realization Γ'' of G whose restriction to G_1 is Γ_1 and whose restriction to G_2 is Γ_2; however, this is one of the four straight-line realizations for which the planarity testing was negative, a contradiction.

It remains to show an $O(n)$-time algorithm that either concludes that G_1 admits no e_{x-1}-outer realization, or determines an e_{x-1}-optimal realization of G_1. A very similar application of the algorithm allows us to either conclude that G_2 admits no e_x-outer realization or to determine an e_x-optimal realization of G_2.

For $i = 1, \ldots, x-1$, let G_1^i be the subgraph of G_1 composed of the 3-cycles $c_1, c_2, \ldots, c_i$; note that $G_1^{x-1} = G_1$. A key observation for our algorithm is the following. Assume that an e_{x-1}-outer realization Γ_1 of G_1 exists; let Γ_1^i be the restriction of Γ_1 to G_1^i. Then Γ_1^i is an e_i-outer realization of G_1^i. This is trivial when $i = x-1$; further, when $i < x-1$, if e_i were not incident to the outer face of Γ_1^i, then e_{x-1} could not be incident to the outer face of Γ_1 without violating the planarity of Γ_1.

Our algorithm works by induction on i in order to either conclude that G_1^i admits no e_i-outer realization (and thus G_1 admits no e_{x-1}-outer realization), or to determine an e_i-optimal realization Γ_1^i of G_1^i. This is trivial when $i = 1$, namely G_1^1 is the 3-cycle c_1 which has a unique planar straight-line realization $\Gamma_1^1 := \mathcal{C}_1$, up to a rigid transformation; then Γ_1^1 is indeed an e_1-optimal realization.

Assume that we have an e_{i-1}-optimal realization Γ_1^{i-1} of G_1^{i-1}, for some $i \in \{2, \ldots, x-1\}$; indeed, if we already concluded that G_1 admits no e_{x-1}-outer realization, there is nothing else to do.

Note that the cycles c_{i-1} and c_i share the edge e_{i-1}; let v_{i-1} and v_i be the vertices of c_{i-1} and c_i, respectively, not incident to e_{i-1}. Any straight-line realization of G_1^i is of one of two types: it is a *different-side* realization if v_{i-1} and v_i lie on different sides of the line through e_{i-1}, while it is a *same-side* realization if v_{i-1} and v_i lie on the same side of the line through e_{i-1}.

In order to determine whether an e_i-outer realization of G_1^i exists, we combine Γ_1^{i-1} with $\mathcal{C}_i$ in two ways, so to form a different-side realization and a same-side realization. Namely, G_1^{i-1} and c_i share the edge e_{i-1}; hence, once Γ_1^{i-1} and $\mathcal{C}_i$ coincide on the representation of e_{i-1}, it still remains to decide whether v_{i-1} and v_i lie on different sides or on the same side of the line through e_{i-1}. This choice leads to two straight-line realizations of G_1^i, which we respectively denote by $\Gamma_1^{i,a}$ and $\Gamma_1^{i,b}$.

We check each of $\Gamma_1^{i,a}$ and $\Gamma_1^{i,b}$ for planarity and discard each realization that is not planar. Further, if $\Gamma_1^{i,b}$ was not discarded, then the triangles $\mathcal{C}_{i-1}$ and $\mathcal{C}_i$ are one nested into the other one (except for the edge e_{i-1}, which is common to both triangles) in such a realization. If $\mathcal{C}_i$ is nested into $\mathcal{C}_{i-1}$, then we discard $\Gamma_1^{i,b}$, as it is not an e_i-outer realization of G_1^i. We have now three cases:

- If we did not discard $\Gamma_1^{i,b}$, then we set $\Gamma_1^i = \Gamma_1^{i,b}$. Indeed, if we did not discard $\Gamma_1^{i,b}$, then $\Gamma_1^{i,b}$ is a planar straight-line realization of G_1^i in which $\mathcal{C}_{i-1}$ is nested into $\mathcal{C}_i$; this implies that all of Γ_1^{i-1} lies inside $\mathcal{C}_i$, hence $\mathcal{I}_{\Gamma_1^i}$ coincides with the closure of the interior of $\mathcal{C}_i$. It follows that Γ_1^i is an e_i-optimal realization of G_1^i, given that in any planar straight-line realization of G_1^i the cycle c_i is represented by the triangle $\mathcal{C}_i$.
- If we discarded $\Gamma_1^{i,b}$ but we did not discard $\Gamma_1^{i,a}$, then we set $\Gamma_1^i = \Gamma_1^{i,a}$. We prove that Γ_1^i is an e_i-optimal realization of G_1^i.

  First, we prove that there exists no same-side e_i-outer realization of G_1^i. This is obvious if $\Gamma_1^{i,b}$ was discarded because $\mathcal{C}_i$ is nested into $\mathcal{C}_{i-1}$, as the same would be true in any same-side realization of G_1^i. Suppose hence that $\Gamma_1^{i,b}$ was discarded because it is not planar and suppose, for a contradiction, that there exists a same-side e_i-outer realization Ψ_1^i of G_1^i. Since Γ_1^{i-1} is e_{i-1}-optimal, it is e_{i-1}-smaller than the restriction of Ψ_1^i to G_1^{i-1}; hence, we can apply Lemma 19

with $H = G_1^{i-1}$, $K = c_i$, $\mathcal{G} = \Psi_1^i$, $e = e_{i-1}$, and $\mathcal{H}' = \Gamma_1^{i-1}$, in order to obtain a planar straight-line realization $\Gamma_1^{i,*}$ of G whose restriction to G_1^{i-1} is Γ_1^{i-1}. Since Ψ_1^i is a same-side realization of G_1^i, we have that $\Gamma_1^{i,*}$ is also a same-side realization of G_1^i. Hence, $\Gamma_1^{i,*}$ is the same realization as $\Gamma_1^{i,b}$, a contradiction to the fact that $\Gamma_1^{i,b}$ is not planar.
Second, since we did not discard $\Gamma_1^{i,a}$, we have that $\Gamma_1^i = \Gamma_1^{i,a}$ is planar. This, together with the fact that Γ_1^i is a different-side realization, implies that Γ_1^i is an e_i-outer realization. The proof that Γ_1^i is an e_i-optimal realization of G_1^i easily follows by induction. Namely, as proved above, any e_i-outer realization of G_1^i is a different-side realization. Consider any such realization Ψ_1^i and let Ψ_1^{i-1} be the restriction of Ψ_1^i to the vertices and edges of G_1^{i-1}. By induction, we have that Γ_1^{i-1} is e_{i-1}-smaller than Ψ_1^{i-1}, hence Γ_1^i is e_i-smaller than Ψ_1^i, given that the realization $\mathcal{C}_i$ of c_i coincides in Γ_1^i and Ψ_1^i.

– Finally, if we discarded both $\Gamma_1^{i,a}$ and $\Gamma_1^{i,b}$, we conclude that there exists no e_i-outer realization of G_1^i. Namely, the proof that there exists no same-side e_i-outer realization of G_1^i is the same as for the case in which we discarded $\Gamma_1^{i,b}$ but we did not discard $\Gamma_1^{i,a}$. The proof that there exists no different-side e_i-outer realization of G_1^i is also very similar to the proof for a same-side realization. Indeed, that we discarded $\Gamma_1^{i,a}$ was unequivocally due to the fact that it is not planar. Suppose, for a contradiction, that there exists a different-side e_i-outer realization Ψ_1^i of G_1^i. Since Γ_1^{i-1} is e_{i-1}-optimal, it is e_{i-1}-smaller than the restriction of Ψ_1^i to G_1^{i-1}; hence, we can apply Lemma 19 with $H = G_1^{i-1}$, $K = c_i$, $\mathcal{G} = \Psi_1^i$, $e = e_{i-1}$, and $\mathcal{H}' = \Gamma_1^{i-1}$, in order to obtain a planar straight-line realization $\Gamma_1^{i,*}$ of G whose restriction to G_1^{i-1} is Γ_1^{i-1}. Since Ψ_1^i is a different-side realization of G_1^i, we have that $\Gamma_1^{i,*}$ is also a different-side realization of G_1^i. Hence, $\Gamma_1^{i,*}$ is the same realization as $\Gamma_1^{i,a}$, a contradiction to the fact that $\Gamma_1^{i,a}$ is not planar.

It remains to show how the described algorithm can be implemented to run in $O(n)$ time. Note that a naive implementation of our algorithm would take $O(n^2)$ time. Indeed, for each of the $O(n)$ inductive steps, the realizations $\Gamma_1^{i,a}$ and $\Gamma_1^{i,b}$, together with their plane embeddings $\mathcal{E}(\Gamma_1^{i,a})$ and $\mathcal{E}(\Gamma_1^{i,b})$, can be constructed in $O(1)$ time from Γ_1^{i-1} and its plane embedding $\mathcal{E}(\Gamma_1^{i-1})$. Further, the test on whether $\mathcal{C}_i$ is nested into $\mathcal{C}_{i-1}$ in $\Gamma_1^{i,b}$ can also be performed in $O(1)$ time. Finally, by Theorem 4, it can be tested in $O(n)$ time whether $\Gamma_1^{i,a}$ and $\Gamma_1^{i,b}$ are planar straight-line realizations of G_1^{i-1} respecting $\mathcal{E}(\Gamma_1^{i,a})$ and $\mathcal{E}(\Gamma_1^{i,b})$, respectively.

In order to achieve $O(n)$ total running time, we do not test planarity at each inductive step. Rather, for $i = 1, \ldots, x-1$, we compute a "candidate" straight-line realization Γ_1^i of G_1^i, and we only test for the planarity of the final realization Γ_1^{x-1}. "Candidate" here means e_i-optimal, however only in the case in which Γ_1^i is planar. Indeed, Γ_1^i might not be planar, in which case the candidate straight-line realization Γ_1^i of G_1^i is not required to satisfy any specific geometric constraint.

More formally, for $i = 1, \ldots, x-1$, we construct a straight-line realization Γ_1^i of G_1^i, together with a plane embedding $\mathcal{E}(\Gamma_1^i)$, satisfying the following invariant: If an e_i-outer realization of G_1^i exists, then Γ_1^i is an e_i-optimal realization of G_1^i and $\mathcal{E}(\Gamma_1^i)$ is the plane embedding of G_1^i corresponding to Γ_1^i. We again remark that, if an e_i-outer realization of G_1^i does not exist, then Γ_1^i is non-planar or the edge e_i is not incident to the outer face of Γ_1^i.

In order to construct Γ_1^i, we maintain:

– the boundary $\mathcal{B}_1^i$ of the convex hull of Γ_1^i (that is, the sequence of vertices along $\mathcal{B}_1^i$);
– a Boolean value $\beta(e_i)$ such that $\beta(e_i) = \texttt{True}$ if and only if e_i is an edge of $\mathcal{B}_1^i$; and
– a distinguished vertex ξ_i of $\mathcal{B}_1^i$ (this information is only available if $\beta(e_i) = \texttt{False}$; the meaning of ξ_i will be explained later).

Our algorithm only guarantees the correctness of the above information in the case in which G_1^i admits an e_i-outer realization. For example, if G_1^i does not admit an e_i-outer realization, $\mathcal{B}_1^i$ might not correctly represent the boundary of the convex hull of Γ_1^i.

Before describing our algorithm, we introduce a problem and a known solution for it. Let $\mathcal{Z}$ be a planar straight-line drawing of a path $Z = (z_1, z_2, \ldots, z_m)$ and, for any $i \in \{1, \ldots, m\}$, let $\mathcal{Z}_i$ be the

restriction of $\mathcal{Z}$ to the subpath $Z_i = (z_1, z_2, \ldots, z_i)$ of Z. Suppose that the locations of $z_1, z_2, \ldots, z_m$ in $\mathcal{Z}$ are unveiled one at a time, in that order. For $i = 1, \ldots, m$, when the location of z_i is unveiled, we want to compute the boundary of the convex hull of $\mathcal{Z}_i$. Melkman [41] presented an algorithm to solve the problem (i.e., to compute the boundaries of the convex hulls of $\mathcal{Z}_1, \mathcal{Z}_2, \ldots, \mathcal{Z}_m$) in total $O(m)$ time.

This result allows us to compute the boundaries $\mathcal{B}_1^1, \mathcal{B}_1^2, \ldots, \mathcal{B}_1^{x-1}$ of the convex hulls of $\Gamma_1^1, \Gamma_1^2, \ldots, \Gamma_1^{x-1}$ in total $O(n)$ time. Indeed, an easy inductive argument proves that G_1^{x-1} contains a path Z spanning all its vertices and such that, for $i = 1, \ldots, x-1$, the vertices of G_1^i induce a subpath Z_i of Z. For $i = 2, \ldots, x-1$, our algorithm constructs Γ_1^i from Γ_1^{i-1} by assigning a position to the vertex v_i of $\mathcal{C}_i$ not incident to e_{i-1}; once we have done that, the algorithm by Melkman [41] can be employed in order to compute $\mathcal{B}_1^i$ from $\mathcal{B}_1^{i-1}$ (more precisely, we run Melkman's algorithm [41] in parallel with our algorithm and feed it the locations of the vertices of Z in the order our algorithm computes them). Note that, if G_1^i admits an e_i-outer realization, then $\mathcal{B}_1^i$ is correctly computed, otherwise $\mathcal{B}_1^i$ might not correctly represent the boundary of the convex hull of Γ_1^i. We will only sketch what Melkman's algorithm [41] does in order to compute $\mathcal{B}_1^1, \mathcal{B}_1^2, \ldots, \mathcal{B}_1^{x-1}$.

The base case takes $O(1)$ time, as G_1^1 has a unique (planar) straight-line realization $\Gamma_1^1 = \mathcal{C}_1$, up to a rigid transformation; further, $\mathcal{E}(\Gamma_1^1)$ is easily computed from Γ_1^1, and we have $\mathcal{B}_1^1 = \mathcal{C}_1$, $\beta(e_1) = \texttt{True}$, and $\xi_1 = \texttt{Null}$. Now assume that, for some $i \in \{2, \ldots, x-1\}$, we have a candidate straight-line realization Γ_1^{i-1} of G_1^{i-1}, together with a plane embedding $\mathcal{E}(\Gamma_1^{i-1})$, satisfying the invariant, the boundary $\mathcal{B}_1^{i-1}$ of the convex hull of Γ_1^{i-1}, the Boolean value $\beta(e_{i-1})$, and the vertex ξ_{i-1}. We are going to distinguish some cases; common to all the cases is the observation that, if Γ_1^{i-1} is not planar, then it is not important how our algorithm computes Γ_1^i, $\mathcal{E}(\Gamma_1^i)$, $\mathcal{B}_1^i$, $\beta(e_i)$, and ξ_i, because, by the invariant, G_1^{i-1} admits no e_{i-1}-outer realization and hence G_1^i admits no e_i-outer realization.

Case 1: $\beta(e_{i-1}) = \texttt{True}$. We check whether Γ_1^{i-1} and $\mathcal{C}_i$ can be combined in order to form a same-side e_i-outer realization of G_1^i (assuming that Γ_1^{i-1} is planar). Since we know $\mathcal{B}_1^{i-1}$, this check only requires $O(1)$ time. Indeed, once $\mathcal{C}_i$ is embedded in such a way that v_{i-1} and v_i lie on the same side of the line through e_{i-1}, it suffices to compare the slopes of the edges of $\mathcal{C}_i$ different from e_{i-1} with the slopes of the segments of $\mathcal{B}_1^{i-1}$ incident to the end-vertices of e_{i-1} and different from e_{i-1}; see Fig. 27(a).

- *Case 1.1:* If the check is successful, we let Γ_1^i be the resulting straight-line realization of G_1^i; further, the plane embedding $\mathcal{E}(\Gamma_1^i)$ can be computed in $O(1)$ time from $\mathcal{E}(\Gamma_1^{i-1})$ by inserting the edges of c_i incident to v_i next to e_{i-1} in the cyclic order of the edges incident to the end-vertices of e_{i-1} and by letting the outer face of $\mathcal{E}(\Gamma_1^i)$ be bounded by the cycle c_i. Finally, we let $\mathcal{B}_1^i = \mathcal{C}_i$, $\beta(e_i) = \texttt{True}$, and $\xi_i = \texttt{Null}$. Note that, if Γ_1^{i-1} is planar, then Γ_1^i is an e_i-optimal realization of G_1^i (this is the realization which was earlier called $\Gamma_1^{i,b}$).
- *Case 1.2:* If the check is unsuccessful, we let Γ_1^i be the straight-line realization obtained from Γ_1^{i-1} by embedding $\mathcal{C}_i$ in such a way that v_{i-1} and v_i lie on different sides of the line through e_{i-1}. Thus, Γ_1^i can be constructed in $O(1)$ time from Γ_1^{i-1}. The plane embedding $\mathcal{E}(\Gamma_1^i)$ can also be computed in $O(1)$ time from $\mathcal{E}(\Gamma_1^{i-1})$; indeed, the set of clockwise orders of the edges incident to each vertex in $\mathcal{E}(\Gamma_1^i)$ is computed as in Case 1.1, while the boundary of the outer face of $\mathcal{E}(\Gamma_1^i)$ is computed from the one of $\mathcal{E}(\Gamma_1^{i-1})$ by letting the two edges of c_i incident to v_i replace the edge e_{i-1}. Note that, if Γ_1^{i-1} is planar, then Γ_1^i is an e_i-optimal realization of G_1^i; this is the realization which was earlier called $\Gamma_1^{i,a}$.
  In order to compute $\mathcal{B}_1^i$ from $\mathcal{B}_1^{i-1}$, Melkman's algorithm [41] works as follows; refer to Fig. 27(b). Let s_{i-1} and t_{i-1} be the end-vertices of e_{i-1} and assume that s_{i-1} immediately precedes t_{i-1} in clockwise order along $\mathcal{B}_1^{i-1}$. Since v_i is a vertex of $\mathcal{B}_1^i$, the algorithm needs to find the vertices of $\mathcal{B}_1^{i-1}$ that follow v_i in clockwise and counter-clockwise order along $\mathcal{B}_1^i$. In order to find the vertex that follows v_i in counter-clockwise order along $\mathcal{B}_1^i$ (the computation of the other vertex is done similarly), the algorithm computes the segments from v_i to the vertices of $\mathcal{B}_1^{i-1}$, starting from s_{i-1} and proceeding in counter-clockwise order along $\mathcal{B}_1^{i-1}$. This computation stops when a segment from v_i to a vertex s'_{i-1} of $\mathcal{B}_1^{i-1}$ is considered such

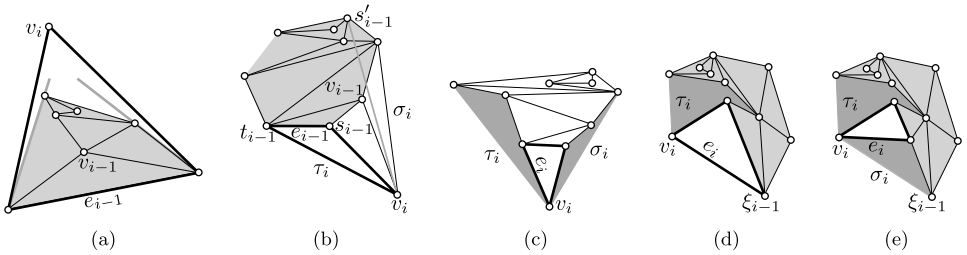

Fig. 27. Embedding $\mathcal{C}_i$ (thick black) together with Γ_1^{i-1}. (a) In order to establish whether Γ_1^{i-1} (whose convex hull is gray) fits inside $\mathcal{C}_i$, the slopes of the edges of $\mathcal{C}_i$ incident to v_i are compared with the slopes of the segments of $\mathcal{B}_1^{i-1}$ incident to the end-vertices of e_{i-1} and different from e_{i-1} (lines through these segments are dark gray). (b) If $\mathcal{C}_i$ is embedded so that v_{i-1} and v_i lie on different sides of the line through e_{i-1}, then $\mathcal{B}_1^i$ has to be computed; Melkman's algorithm [41] examines the segments from v_i to the vertices that follow s_{i-1} in counter-clockwise order along $\mathcal{B}_1^{i-1}$, until it finds a segment (dark gray) from v_i to a vertex s'_{i-1} that keeps the previously encountered segment to the right. (c) If e_i lies in the interior of the convex hull of Γ_1^i, except possibly for its end-vertices, then the segments σ_i and τ_i of $\mathcal{B}_1^i$ incident to $\xi_i = v_i$ represent the only "way out" of $\mathcal{B}_1^i$ for a planar straight-line realization of G_1^{x-1}; in particular, in any e_{x-1}-outer realization of G_1^{x-1}, the vertices and edges that are not in G_1^i and that lie inside $\mathcal{B}_1^i$ can only lie in the dark gray regions incident to σ_i and τ_i. (d) Illustration for the case in which v_i lies outside the convex hull of Γ_1^{i-1} and ξ_{i-1} is the end-vertex of e_i different from v_i. (e) Illustration for the case in which v_i lies outside the convex hull of Γ_1^{i-1} and ξ_{i-1} is not the end-vertex of e_i different from v_i. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

that the oriented line that passes first through v_i and then through s'_{i-1} keeps the previously encountered segment s to the right. When that happens, the end-vertex of s different from v_i is the vertex that follows v_i in counter-clockwise order along $\mathcal{B}_1^i$. This computation does not, in general, take $O(1)$ time, however it takes linear time in the number of vertices of $\mathcal{B}_1^{i-1}$ that are not in $\mathcal{B}_1^i$, hence it takes $O(n)$ time over all the convex hull computations.

Let σ_i and τ_i be the segments of $\mathcal{B}_1^i$ incident to v_i. In order to compute $\beta(e_i)$ and ξ_i, we check whether e_i coincides with one of σ_i and τ_i. In the positive case, since e_i is an edge of $\mathcal{B}_1^i$, we let $\beta(e_i) =$ `True` and $\xi_i =$ `Null`. In the negative case, since e_i is not an edge of $\mathcal{B}_1^i$, we let $\beta(e_i) =$ `False` and $\xi_i = v_i$. We are now ready to explain the meaning of the information conveyed by $\beta(e_i)$ and ξ_i; refer to Fig. 27(c). If e_i lies in the interior of the convex hull of Γ_1^i, except possibly for one of its end-vertices, then the segments σ_i and τ_i of $\mathcal{B}_1^i$ incident to ξ_i represent the only "way out" of $\mathcal{B}_1^i$ for a planar straight-line realization of G_1^{x-1}. Formally, let j be the smallest index such that $j > i$ and such that v_j lies outside $\mathcal{B}_1^i$ in a planar straight-line realization of G_1^j, if any such an index exists; then (at least) one of the two edges incident to v_j in G_1^j crosses σ_i or τ_i. This very same property is exploited in Melkman's algorithm [41], and comes from the fact that, by the connectivity of G_1^i, any path that starts at v_i, that does not share any vertex with G_1^i other than v_i, and that first intersects the interior of the convex hull of Γ_1^i and then crosses $\mathcal{B}_1^i$ in a segment different from σ_i and τ_i crosses Γ_1^i as well.

Case 2: $\beta(e_{i-1}) =$ `False`. In this case Γ_1^{i-1} lies on both sides of the line through e_{i-1}, hence Γ_1^{i-1} and $\mathcal{C}_i$ cannot be combined in order to form a same-side e_i-outer realization of G_1^i, even if Γ_1^{i-1} is planar. Thus, as in Case 1.2, we let Γ_1^i be the straight-line realization obtained from Γ_1^{i-1} by embedding $\mathcal{C}_i$ in such a way that v_{i-1} and v_i lie on different sides of the line through e_{i-1}; this is again the realization which was earlier called $\Gamma_1^{i,a}$. The plane embedding $\mathcal{E}(\Gamma_1^i)$ is computed in $O(1)$ time from $\mathcal{E}(\Gamma_1^{i-1})$ as in Case 1.2. Then Γ_1^i satisfies the invariant; indeed, in the description of the algorithm, it was proved that, if $\Gamma_1^{i,a}$ is planar, then it is e_i-optimal, otherwise G_1^i admits no e_i-outer realization. Further, Γ_1^i can be constructed in $O(1)$ time from Γ_1^{i-1}. Note that, in this case, it might happen that Γ_1^i is not planar even if Γ_1^{i-1} is planar; however, we do not check the planarity of Γ_1^i at this time.

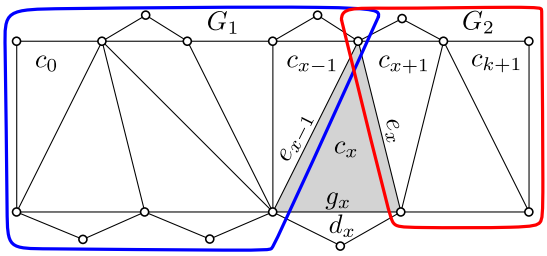

Fig. 28. The outerplane embedding of an outerpillar. The gray face is the one bounded by c_x. The graphs G_1 and G_2 are enclosed inside fat curves.

In order to compute $\mathcal{B}_1^i$, $\beta(e_i)$, and ξ_i, we exploit the information stored in ξ_{i-1}. As explained in Case 1.2, if Γ_1^{i-1} is planar, the value ξ_{i-1} represents a vertex on the boundary $\mathcal{B}_1^{i-1}$ of the convex hull of Γ_1^{i-1} such that v_i lies outside $\mathcal{B}_1^{i-1}$ only if the (at least one) edge e of c_i incident to v_i and not incident to ξ_{i-1} crosses σ_{i-1} or τ_{i-1}, where σ_{i-1} and τ_{i-1} are the segments incident to ξ_{i-1} in $\mathcal{B}_1^{i-1}$. Hence, we (as in Melkman's algorithm [41]) check in $O(1)$ time whether e crosses σ_{i-1} or τ_{i-1}.

- In the positive case, v_i lies outside the convex hull of Γ_1^{i-1}; then $\mathcal{B}_1^i$ is constructed as in Case 1.2, where the segment between σ_i and τ_i that crosses e plays the role of e_{i-1}. If ξ_{i-1} is the end-vertex of e_i different from v_i, as in Fig. 27(d), then we set $\beta(e_i) = \texttt{True}$ and $\xi_i = \texttt{Null}$ (in this case G_1^i "made it out" of $\mathcal{B}_1^{i-1}$), otherwise, as in Fig. 27(e), we set $\beta(e_i) = \texttt{False}$ and $\xi_i = v_i$.
- In the negative case, v_i lies inside the convex hull of Γ_1^{i-1} or Γ_1^i is not planar (or both). Hence, we set $\mathcal{B}_1^i = \mathcal{B}_1^{i-1}$, $\beta(e_i) = \texttt{False}$, and $\xi_i = \xi_{i-1}$.

This concludes the description of the inductive case of our algorithm. Each inductive step takes $O(1)$ time, except for the computation of the boundaries $\mathcal{B}_1^1, \mathcal{B}_1^2, \ldots, \mathcal{B}_1^{x-1}$, which however takes total $O(n)$ time as explained above. Hence, the overall running time of the algorithm is in $O(n)$. □

Proof of Theorem 14. The proof uses several ideas from the proof of Theorem 13. We hence omit details of arguments that have been already detailed in that proof.

Let $G = (V, E, \lambda)$ be an n-vertex weighted maximal outerpillar (as the one in Fig. 28) and let $\mathcal{O}$ be its outerplane embedding, which can be found in $O(n)$ time [25,42,54]. Since G is an outerpillar, its dual tree is a caterpillar T, whose spine we denote by $P := (p_1, \ldots, p_k)$; note that T can be easily recovered from $\mathcal{O}$ in $O(n)$ time. We arbitrarily select two leaves of T, respectively adjacent to p_1 and p_k, and denote them by p_0 and p_{k+1}; we further denote by P' the path $(p_0, p_1, \ldots, p_k, p_{k+1})$. Each node p_i of P' corresponds to an internal face of $\mathcal{O}$; we denote by c_i the 3-cycle of G bounding such a face. Further, each node p_i in P' is adjacent to at most one leaf not in P' (the satisfaction of this property was the reason to augment P to P'), which corresponds to an internal face of $\mathcal{O}$; we denote by d_i the 3-cycle of G bounding such a face. Each cycle c_i has a unique (planar) straight-line realization (up to a rigid transformation), which we denote by $\mathcal{C}_i$, and each cycle d_i has a unique (planar) straight-line realization (up to a rigid transformation), which we denote by $\mathcal{D}_i$. For $i = 0, \ldots, k$, let e_i be the edge of G that is dual to the edge (p_i, p_{i+1}) of P'; note that e_i is an internal edge of $\mathcal{O}$. Further, for every index $i \in \{1, \ldots, k\}$ such that d_i exists, let g_i be the edge shared by c_i and d_i.

Let c^* be a 3-cycle of G whose area is maximum among all the 3-cycles of G. Let $x \in \{0, 1, \ldots, k+1\}$ be the index such that we have either $c^* = c_x$ or $c^* = d_x$. Let G_1 be the subgraph of G composed of the 3-cycles c_i and of the 3-cycles d_i with $i = 0, \ldots, x-1$ (some of the 3-cycles d_i might not exist). Analogously, let G_2 be the subgraph of G composed of the 3-cycles c_i and of the 3-cycles d_i with $i = x+1, \ldots, k+1$ (again, some of the 3-cycles d_i might not exist). One of these graphs might be undefined if $x = 0$ or $x = k+1$; however, in the following, we assume that $1 \leq x \leq k$ as the arguments for the cases $x = 0$ and $x = k+1$ are analogous and actually simpler. Note that G_1 and

G_2 are maximal outerpillars. Also, we assume that the 3-cycle d_x exists, as the argument for the case in which it does not is simpler.

Assume that a planar straight-line realization Γ of G exists; for $i = 1, 2$, let Γ_i be the restriction of Γ to G_i. If $c^* = c_x$, then c_x does not lie inside any 3-cycle of G in Γ. If $c^* = d_x$, then c_x might lie inside d_x in Γ, however it does not lie inside any other 3-cycle of G in Γ, as otherwise, by planarity, such a 3-cycle would also contain d_x, which is not possible. Therefore, both if $c^* = c_x$ and if $c^* = d_x$, we have that Γ_1 is an e_{x-1}-outer realization of G_1 and that Γ_2 is an e_x-outer realization of G_2.

We show an $O(n^3)$-time algorithm that either concludes that G_1 admits no e_{x-1}-outer realization or constructs an e_{x-1}-optimal set $\mathcal{R}_1$ of e_{x-1}-outer realizations of G_1, where for each realization Γ_1 in $\mathcal{R}_1$ the plane embedding $\mathcal{E}(\Gamma_1)$ is constructed as well. The same algorithm is then also used in order to either conclude that G_2 admits no e_x-outer realization or to construct an e_x-optimal set $\mathcal{R}_2$ of e_x-outer realizations of G_2, where for each realization Γ_2 in $\mathcal{R}_2$ the plane embedding $\mathcal{E}(\Gamma_2)$ is constructed as well.

For $i = 0, 1, \ldots, x-1$, let G_1^i be the subgraph of G_1 composed of the 3-cycles $c_0, c_1, \ldots, c_i$ and of the 3-cycles $d_0, d_1, \ldots, d_i$ (some of these 3-cycles might not exist); note that $G_1^{x-1} = G_1$. As in the proof of Theorem 13, we use the observation that the restriction of any e_{x-1}-outer realization of G_1 to G_1^i is an e_i-outer realization of G_1^i.

Our algorithm works by induction on i in order to either conclude that G_1^i admits no e_i-outer realization (and thus G_1 admits no e_{x-1}-outer realization), or to determine an e_i-optimal set $\mathcal{R}_1^i$ of e_i-outer realizations of G_1^i with $|\mathcal{R}_1^i| \leq i+1$, where for each realization Γ_1^i in $\mathcal{R}_1^i$ the plane $\mathcal{E}(\Gamma_1^i)$ is constructed as well. This is trivial when $i = 0$, namely G_1^0 is the 3-cycle c_0 which has a unique planar straight-line realization $\Gamma_1^0 := \mathcal{C}_0$, up to a rigid transformation; then $\mathcal{R}_1^0 := \{\Gamma_1^0\}$ is indeed an e_0-optimal set of e_0-outer realizations of G_1^0 with $|\mathcal{R}_1^0| \leq 1$ and $\mathcal{E}(\Gamma_1^0)$ is easily computed from Γ_1^0.

Assume that we have an e_{i-1}-optimal set $\mathcal{R}_1^{i-1}$ of e_{i-1}-outer realizations of G_1^{i-1} with $|\mathcal{R}_1^{i-1}| \leq i$, for some $i \in \{1, \ldots, x-1\}$, and we have, for each realization Γ_1^{i-1} in $\mathcal{R}_1^{i-1}$, the plane embedding $\mathcal{E}(\Gamma_1^{i-1})$; indeed, if we already concluded that G_1 admits no e_{x-1}-outer realization, there is nothing else to do. We present an algorithm that either concludes that G_1 admits no e_{x-1}-outer realization or constructs the set $\mathcal{R}_1^i$ and, for each realization Γ_1^i in $\mathcal{R}_1^i$, the plane embedding $\mathcal{E}(\Gamma_1^i)$. The algorithm consists of three steps. Step 1 tries to construct an e_i-outer realization of G_1^i in which both G_1^{i-1} and d_i lie inside c_i, as in Fig. 29(a). If this construction succeeds, then $\mathcal{R}_1^i$ consists uniquely of such a realization of G_1^i and Steps 2 and 3 are not performed. Otherwise, Steps 2 and 3 try to construct e_i-outer realizations of G_1^i in which G_1^{i-1} and/or d_i lie outside c_i, as in Figs. 29(b) and 29(c). In particular, Step 2 considers the realizations in which G_1^{i-1} lies inside c_i and d_i lies outside c_i. This step adds to $\mathcal{R}_1^i$ at most one of such realizations. Step 3 instead considers the realizations in which G_1^{i-1} lies outside c_i and d_i lies inside c_i (if it fits) or outside c_i (otherwise). This step adds to $\mathcal{R}_1^i$ every constructed realization of G_1^i. In all three steps, the used realizations of G_1^{i-1} are those in the inductively constructed set $\mathcal{R}_1^{i-1}$. We next describe each step in detail.

- **Step 1.** In the first step, we aim for the construction of an e_i-outer realization of G_1^i in which both G_1^{i-1} and d_i lie inside c_i (except for the edges e_{i-1} and g_i, which c_i shares with G_1^{i-1} and d_i, respectively), as the one in Fig. 29(a). This is done by checking whether any of the $O(n)$ realizations of G_1^{i-1} in $\mathcal{R}_1^{i-1}$ fits inside $\mathcal{C}_i$ together with $\mathcal{D}_i$. More precisely, we act as follows.

  (A) We test in $O(1)$ time whether $\mathcal{C}_{i-1}$ lies inside $\mathcal{C}_i$ once they coincide on the representation of e_{i-1} and once their vertices not incident to e_{i-1} lie on the same side of the line through e_{i-1}. If the test is positive, we proceed with the first step of the algorithm, otherwise we conclude that no realization of G_1^{i-1} fits inside $\mathcal{C}_i$, we terminate the first step of the algorithm and proceed to the second one.

  (B) We test in $O(1)$ time whether $\mathcal{D}_i$ lies inside $\mathcal{C}_i$ once they coincide on the representation of g_i and once their vertices not incident to g_i lie on the same side of the line through g_i; if d_i does not exist, this test is skipped. If the test is positive, we proceed with the first step of the algorithm, otherwise we conclude that $\mathcal{D}_i$ does not fit inside $\mathcal{C}_i$, we terminate the first step of the algorithm and proceed to the second one.

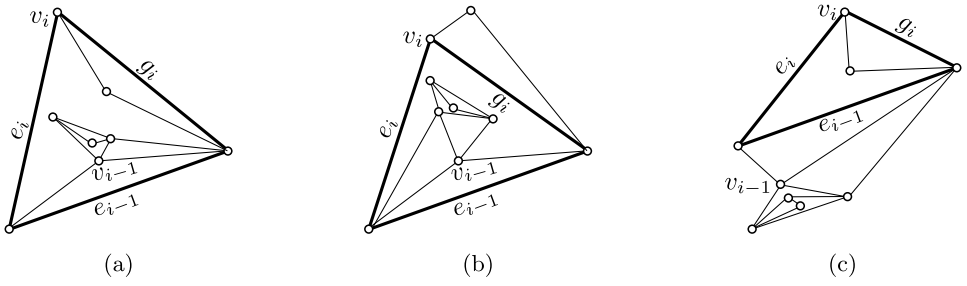

Fig. 29. Three e_i-outer realizations of G_1^i. The cycle c_i is represented by a thick line. (a) G_1^{i-1} and d_i lie inside c_i; (b) G_1^{i-1} lies inside c_i and d_i lies outside c_i; (c) G_1^{i-1} lies outside c_i.

(C) For each realization Γ_1^{i-1} of G_1^{i-1} in $\mathcal{R}_1^{i-1}$, we construct a straight-line realization Γ_1^i of G_1^i in which G_1^{i-1} is represented by Γ_1^{i-1} and $\mathcal{C}_{i-1}$ and $\mathcal{D}_i$ are embedded inside $\mathcal{C}_i$. Such a realization can be obtained in $O(1)$ time by augmenting Γ_1^{i-1} with a representation of the edges of c_i and d_i not in G_1^{i-1}. Further, the plane embedding $\mathcal{E}(\Gamma_1^i)$ can be constructed in $O(1)$ time by inserting the edges incident to v_i next to e_{i-1} in the cyclic order of the edges incident to the end-vertices of e_{i-1}, by then inserting the edges of d_i different from g_i next to g_i in the cyclic order of the edges incident to the end-vertices of g_i, and by letting the outer face of $\mathcal{E}(\Gamma_1^i)$ be bounded by the cycle c_i. We then test in $O(n)$ time, by means of Theorem 4, whether Γ_1^i is a planar straight-line realization of G_1^i respecting $\mathcal{E}(\Gamma_1^i)$.

The total running time of the first step of the algorithm is hence in $O(n^2)$. If (at least) one of these tests succeeds in constructing a realization Γ_1^i, then we complete the induction with $\mathcal{R}_1^i = \{\Gamma_1^i\}$. If none of the tests succeeds, we proceed to the second step of the algorithm.

- **Step 2.** In the second step, we aim for the construction of an e_i-outer realization of G_1^i in which G_1^{i-1} lies inside c_i (except for the edge e_{i-1}, which is shared by G_1^{i-1} and c_i) and d_i lies outside c_i (except for the edge g_i, which is shared by c_i and d_i), as in Fig. 29(b). If d_i does not exist, then this step is actually ignored, given that the desired representation of G_1^i would have been already constructed in the first step. As in the first step, we first check whether $\mathcal{C}_{i-1}$ lies inside $\mathcal{C}_i$ once they coincide on the representation of e_{i-1} and once their vertices not incident to e_{i-1} lie on the same side of the line through e_{i-1}. If the test is negative, we conclude that no realization of G_1^{i-1} fits inside $\mathcal{C}_i$, we terminate the second step of the algorithm and proceed to the third one with $\mathcal{R}_1^i = \emptyset$. If the test is positive, we independently consider each of the $O(n)$ realizations of G_1^{i-1} in $\mathcal{R}_1^{i-1}$, say Γ_1^{i-1}. We augment Γ_1^{i-1} to a realization Γ_1^i of G_1^i (and compute the plane embedding $\mathcal{E}(\Gamma_1^i)$ from the plane embedding $\mathcal{E}(\Gamma_1^{i-1})$) in $O(1)$ time similarly as in the first step, except that $\mathcal{D}_i$ is now embedded outside $\mathcal{C}_i$ and hence the outer face of $\mathcal{E}(\Gamma_1^i)$ is now delimited by the edges of c_i and d_i different from g_i. By means of Theorem 4, we test in $O(n)$ time whether Γ_1^i is a planar straight-line realization of G_1^i respecting $\mathcal{E}(\Gamma_1^i)$. Hence, the total running time of the second step is in $O(n^2)$. If (at least) one of these tests shows that a realization Γ_1^i of G_1^i is planar, we conclude this step by initializing $\mathcal{R}_1^i = \{\Gamma_1^i\}$. If none of the tests succeeds, then we initialize $\mathcal{R}_1^i = \emptyset$. In both cases, we proceed to the third step of the algorithm.
- **Step 3.** In the third step, we aim for the construction of (possibly multiple) e_i-outer realizations of G_1^i in which G_1^{i-1} lies outside c_i (except for the edge e_{i-1}, which is shared by G_1^{i-1} and c_i), as in Fig. 29(c). This is done as follows. Preliminarly, we construct a realization of $c_i \cup d_i$ (if d_i exists). In order to do that, we check in $O(1)$ time whether $\mathcal{D}_i$ fits inside $\mathcal{C}_i$; in the positive case, we place it inside $\mathcal{C}_i$, otherwise we place it outside $\mathcal{C}_i$. We now independently consider each e_{i-1}-outer realization Γ_1^{i-1} of G_1^{i-1} in $\mathcal{R}_1^{i-1}$ and we put it together with the constructed

representation of $c_i \cup d_i$ (if d_i does not exist, we just put Γ_1^{i-1} together with $\mathcal{C}_i$), so that the vertices of c_{i-1} and c_i not incident to e_{i-1} lie on different sides of the line through e_{i-1}. This can be done in $O(1)$ time, as in the first and second steps, by augmenting Γ_1^{i-1} with a representation of the edges of c_i and d_i not in G_1^{i-1}. The plane embedding of the constructed representation of G_1^i can also be constructed in $O(1)$ time from the plane embedding $\mathcal{E}(\Gamma_1^{i-1})$, similarly as in the first and second steps. We now check in $O(n)$ time whether the constructed representation of G_1^i is a planar straight-line realization of G_1^i respecting the constructed plane embedding, by means of Theorem 4; in the positive case, we add it to $\mathcal{R}_1^i$. Hence, the total running time of the third step is in $O(n^2)$. After the third step, if $\mathcal{R}_1^i = \emptyset$, we conclude that G_1 admits no e_{x-1}-outer realization.

This concludes the description of the algorithm. We now prove its correctness.

First, we prove that $|\mathcal{R}_1^i| \leq i+1$. This is obvious if the first step succeeded in the construction of a realization of G_1^i, as in that case we have $|\mathcal{R}_1^i| = 1 < i+1$. Otherwise, at most one realization of G_1^i was inserted into $\mathcal{R}_1^i$ in the second step; further, during the third step, at most one realization of G_1^i was inserted into $\mathcal{R}_1^i$ for each realization in $\mathcal{R}_1^{i-1}$. Since $|\mathcal{R}_1^{i-1}| \leq i$, by the inductive hypothesis, it follows that $|\mathcal{R}_1^i| \leq i+1$.

Second, we have that each realization in $\mathcal{R}_1^i$ is planar, as ensured by a test performed on it before the insertion into $\mathcal{R}_1^i$. Further, each realization in $\mathcal{R}_1^i$ is e_i-outer. Namely, if the vertices of c_{i-1} and c_i that are not incident to e_{i-1} lie on the same side of the line through e_{i-1}, we test whether $\mathcal{C}_{i-1}$ lies inside $\mathcal{C}_i$; further, if the vertices of d_i and c_i that are not incident to g_i lie on the same side of the line through g_i, we test whether $\mathcal{D}_i$ lies inside $\mathcal{C}_i$. A realization is added to $\mathcal{R}_1^i$ only if both tests succeed, which ensures that $\mathcal{C}_i$ does not lie inside $\mathcal{C}_{i-1}$ or $\mathcal{D}_i$ in any realization in $\mathcal{R}_1^i$. A containment of $\mathcal{C}_i$ inside any other 3-cycle of G_1^i would contradict the planarity of the corresponding realization.

Finally, we prove that $\mathcal{R}_1^i$ is e_i-optimal. This can be done very similarly as in the proof of Theorem 13. Namely, consider any e_i-outer realization Ψ_1^i of G_1^i and let Ψ_1^{i-1} be the restriction of Ψ_1^i to G_1^{i-1}. Since $\mathcal{R}_1^{i-1}$ is e_{i-1}-optimal, there exists an e_{i-1}-outer realization Λ_1^{i-1} of G_1^{i-1} in $\mathcal{R}_1^{i-1}$ such that Λ_1^{i-1} is e_{i-1}-smaller than Ψ_1^{i-1}. Hence, we can apply Lemma 19 with $H = G_1^{i-1}$, $K = c_i \cup d_i$ (or $K = c_i$ if d_i does not exist), $\mathcal{G} = \Psi_1^i$, $e = e_{i-1}$, and $\mathcal{H}' = \Lambda_1^{i-1}$, in order to obtain an e_i-outer realization Λ_1^i of G_1^i whose restriction to G_1^{i-1} can be obtained by a rigid transformation of Λ_1^{i-1} and such that $\mathcal{C}_{i-1}$ lies inside $\mathcal{C}_i$ in Λ_1^i if and only if $\mathcal{C}_{i-1}$ lies inside $\mathcal{C}_i$ in Ψ_1^i. It only remains to prove that Λ_1^i, or an e_i-outer realization that is e_i-smaller than Λ_1^i, actually belongs to $\mathcal{R}_1^i$.

First, if $\mathcal{R}_1^i$ contains a single realization Γ_1^i added during the first step, then the outer face of Γ_1^i is bounded by $\mathcal{C}_i$; since the set $\mathcal{I}_{\Lambda_1^i}$ contains at least the closure of the interior of $\mathcal{C}_i$, we have that Γ_1^i is e_i-smaller than Λ_1^i. We can hence assume that Λ_1^{i-1} lies outside $\mathcal{C}_i$ or that $\mathcal{D}_i$ lies outside $\mathcal{C}_i$ (or both). Indeed, if both Λ_1^{i-1} and $\mathcal{D}_i$ lied inside $\mathcal{C}_i$, then the first step of the algorithm would have constructed Λ_1^i, and hence $\mathcal{R}_1^i$ would contain a single realization Γ_1^i added during the first step (note that Γ_1^i would not necessarily be the same as Λ_1^i, however the outer faces of both Γ_1^i and Λ_1^i would be delimited by $\mathcal{C}_i$).

Second, consider the case in which Λ_1^{i-1} lies inside $\mathcal{C}_i$ and $\mathcal{D}_i$ lies outside $\mathcal{C}_i$ in Λ_1^i. This implies that the second step of the algorithm constructs Λ_1^i, and hence $\mathcal{R}_1^i$ contains a realization Γ_1^i added during the second step of the algorithm. The outer face of Γ_1^i is bounded by the edges of $\mathcal{C}_i$ and $\mathcal{D}_i$ different from g_i, same as Λ_1^i, hence Γ_1^i is e_i-smaller than Λ_1^i.

Third, consider the case in which Λ_1^{i-1} lies outside $\mathcal{C}_i$. If $\mathcal{D}_i$ lies inside $\mathcal{C}_i$ in Λ_1^i, or if $\mathcal{D}_i$ lies outside $\mathcal{C}_i$ in Λ_1^i and does not fit inside $\mathcal{C}_i$, then the third step of the algorithm constructs Λ_1^i and adds it to $\mathcal{R}_1^i$. If $\mathcal{D}_i$ lies outside $\mathcal{C}_i$ in Λ_1^i and fits inside $\mathcal{C}_i$, then the third step of the algorithm constructs a realization Γ_1^i which coincides with Λ_1^i when restricted to $G_1^{i-1} \cup c_i$, but in which $\mathcal{D}_i$ lies inside $\mathcal{C}_i$. Then we have that Γ_1^i is e_i-smaller than Λ_1^i. This concludes the proof of the e_i-optimality of $\mathcal{R}_1^i$ and of the correctness of the inductive step.

The inductive step is hence performed in $O(n^2)$ time. Since there are $O(n)$ inductive steps, our algorithm takes $O(n^3)$ time in order to either conclude that G_1 admits no e_{x-1}-outer realization or to construct an e_{x-1}-optimal set $\mathcal{R}_1$ with $|\mathcal{R}_1| \in O(n)$ of e_{x-1}-outer realizations of G_1 together with the plane embedding $\mathcal{E}(\Gamma_1)$ of each realization Γ_1 in $\mathcal{R}_1$; the algorithm also takes $O(n^3)$ time in

order to either conclude that G_2 admits no e_x-outer realization or to construct an e_x-optimal set $\mathcal{R}_2$ with $|\mathcal{R}_2| \in O(n)$ of e_x-outer realizations of G_2 together with the plane embedding $\mathcal{E}(\Gamma_2)$ of each realization Γ_2 in $\mathcal{R}_2$.

As argued above, if G_1 admits no e_{x-1}-outer realization or G_2 admits no e_x-outer realization, then G admits no planar straight-line realization. Assume hence that $\mathcal{R}_1$ and $\mathcal{R}_2$ are both non-empty. We independently consider each pair (Γ_1, Γ_2) such that $\Gamma_1 \in \mathcal{R}_1$ and $\Gamma_2 \in \mathcal{R}_2$; since $|\mathcal{R}_1|$ and $|\mathcal{R}_2|$ are both in $O(n)$, there are $O(n^2)$ such pairs. We construct eight straight-line realizations of G, each with the corresponding plane embedding, according to the following choices:

- the vertices of c_{x-1} and c_x not incident to e_{x-1} might lie on the same side or on different sides of the line through e_{x-1};
- the vertices of c_{x+1} and c_x not incident to e_x might lie on the same side or on different sides of the line through e_x; and
- the vertices of d_x and c_x not incident to g_x might lie on the same side or on different sides of the line through g_x.

We test in $O(n)$ time whether each of these straight-line realizations is planar and respects the corresponding plane embedding, by means of Theorem 4. Hence, all these tests can be performed in overall $O(n^3)$ time. If (at least) one of these realizations is planar, then it is the desired planar straight-line realization of G. Otherwise, there exists no planar straight-line realization of G; this can be proved as in the proof of Theorem 13. Namely, if a planar straight-line realization Γ of G exists, then the restriction of Γ to G_1 and the restriction of Γ to G_2 can be replaced by straight-line realizations obtained by rigid transformations of suitable realizations in $\mathcal{R}_1$ and $\mathcal{R}_2$, respectively. By Lemma 19, this results in a planar straight-line realization of G which, by construction, is one of the straight-line realizations for which the planarity testing was negative. This contradiction concludes the proof of the theorem. □ □

7. 2-Trees with short longest path

In this section, we show that the FEPR problem is polynomial-time solvable for weighted 2-trees whose longest path has bounded length, where the *length* of a path is the number of edges in it. Our algorithm actually runs in polynomial time in a more general case: if the height of the SPQ-tree of the 2-tree is bounded. We start this section by introducing the necessary definitions and notation.

The notion of *SPQ-tree* is a specialization of the well-known notion of *SPQR-tree* [9,33] for *series–parallel graphs* [8,53]. A series–parallel graph is any graph that can be obtained by means of the following recursive construction (see Fig. 30).

- An edge (u, v) is a series–parallel graph with two poles u and v; its SPQ-tree $\mathcal{T}$ is rooted tree with a single *Q-node*. We say that (u, v) *corresponds* to the Q-node (and vice versa).
- Let $G_1, \ldots, G_k$ be series–parallel graphs, where G_i has poles u_i and v_i, for $i = 1, \ldots, k$; also, let $\mathcal{T}_1, \ldots, \mathcal{T}_k$ be their SPQ-trees. The graph obtained by identifying the vertices $u_1, \ldots, u_k$ into a single vertex u and the vertices $v_1, \ldots, v_k$ into a single vertex v is a series–parallel graph G with poles u and v. The SPQ-tree $\mathcal{T}$ of G is the tree whose root is a *P-node* which is the parent of the roots of $\mathcal{T}_1, \ldots, \mathcal{T}_k$. We say that G *corresponds* to $\mathcal{T}$.
- Let $G_1, \ldots, G_k$ be series–parallel graphs, where G_i has poles u_i and v_i, for $i = 1, \ldots, k$; also, let $\mathcal{T}_1, \ldots, \mathcal{T}_k$ be their SPQ-trees. The graph obtained by identifying the vertex v_i with the vertex u_{i+1}, for $i = 1, \ldots, k-1$, is a series–parallel graph G with poles u_1 and v_k. The SPQ-tree $\mathcal{T}$ of G is the tree whose root is an *S-node* which is the parent of the roots of $\mathcal{T}_1, \ldots, \mathcal{T}_k$. We say that G *corresponds* to $\mathcal{T}$.

Every 2-tree G is a series–parallel graph (see, e.g., [17, Chapter 5]) such that:

(C1) The root of the SPQ-tree $\mathcal{T}$ of G is a P-node.
(C2) Each P-node has at least two children; one of them is a Q-node and the others are S-nodes.
(C3) Each S-node has exactly two children, each of which is either a Q-node or a P-node.

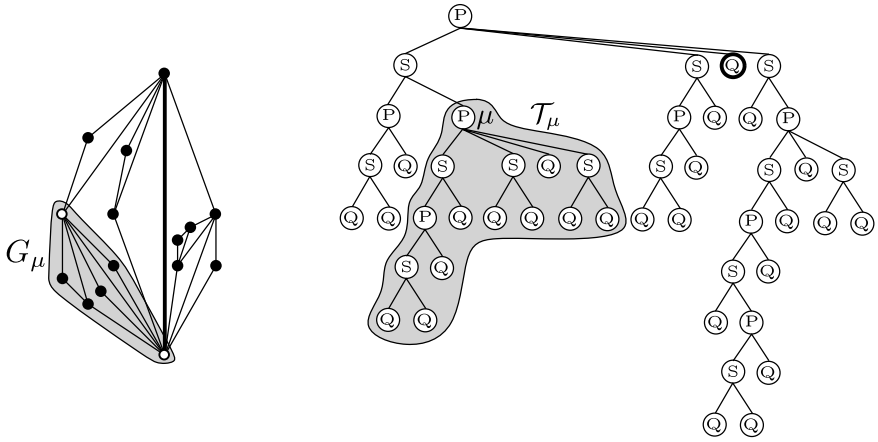

Fig. 30. A 2-tree G and its SPQ-tree $\mathcal{T}$. The gray regions show G_μ and $\mathcal{T}_\mu$, for a node μ of $\mathcal{T}$. The poles of μ are represented by empty disks in G. The fat edge corresponds to the Q-node that is a child of the root of $\mathcal{T}$. The height of $\mathcal{T}$ is 8, the one of $\mathcal{T}_\mu$ is 4.

Moreover, if e is a prescribed edge of G, then the SPQ-tree of G can be constructed in such a way that the Q-node corresponding to e is a child of the root ρ of the SPQ-tree (that is, e is the edge between the poles of ρ). We will use this property later.

Let G be a 2-tree, $\mathcal{T}$ be the SPQ-tree of G, and μ be a node of $\mathcal{T}$. We denote by u_μ and v_μ the poles of μ, by $\mathcal{T}_\mu$ the subtree of $\mathcal{T}$ rooted at μ, and by G_μ the subgraph of G corresponding to $\mathcal{T}_\mu$. The *height* of $\mathcal{T}$ is the maximum number of edges in any root-to-leaf path in $\mathcal{T}$. By Conditions (C1)–(C2), the height of $\mathcal{T}$ is even and positive.

We can upper bound the height of an SPQ-tree of a 2-tree G in terms of the length of the longest path of G, as proved in the following lemma.

Lemma 20. *Let G be a 2-tree, let ℓ be the length of the longest path of G, let $\mathcal{T}$ be the SPQ-tree of G, and let h be the height of $\mathcal{T}$. Then we have $h \leq 2\ell - 2$.*

Proof. We prove the following claim. For every P-node μ in $\mathcal{T}$, there exists a path P_μ in G_μ that connects the poles u and v of μ and whose length is $\frac{h_\mu}{2} + 1$, where h_μ is the height of $\mathcal{T}_\mu$. The claim implies the lemma by choosing μ to be the root of $\mathcal{T}$; this can be done since, by Condition (C1), the root of $\mathcal{T}$ is a P-node.

We prove the claim by induction on h_μ. Recall that h_μ is even and positive. Let ν be a child of μ such that the height of $\mathcal{T}_\nu$ is $h_\mu - 1$; by Condition (C2) and since $h_\mu \geq 2$, we have that ν is an S-node with poles u and v. By Condition (C3), we have that ν has two children ν_1 and ν_2, each of which is either a P-node or a Q-node. Assume, w.l.o.g. up to a relabeling, that u is a pole of ν_1, that v is a pole of ν_2, and that w is the pole shared by ν_1 and ν_2.

In the base case, we have $h_\mu = 2$. Then ν_1 and ν_2 are both Q-nodes, as otherwise the height of $\mathcal{T}$ would be greater than 2. It follows that G contains the path $P_\mu = (u, w, v)$.

In the inductive case, we have $h_\mu > 2$. Assume that the height of $\mathcal{T}_{\nu_1}$ is $h_\mu - 2$, as the case in which the height of $\mathcal{T}_{\nu_2}$ is $h_\mu - 2$ is analogous. Since $h_\mu > 2$, we have that ν_1 is a P-node. By induction, G_{ν_1} contains a path $\mathcal{P}_{\nu_1}$ with length $\frac{h_\mu - 2}{2} + 1$ between u and w. Further, G_{ν_2} contains the edge (w, v); this is obvious if ν_2 is a Q-node and comes from Condition (C2) otherwise. The path $\mathcal{P}_{\nu_1} \cup (w, v)$ is the desired path $\mathcal{P}_\mu$ with length $\frac{h_\mu}{2} + 1$ between u and v. This completes the induction and hence the proof of the lemma. □

The main result of this section is the following.

Theorem 15. *Let G an n-vertex weighted 2-tree, let e be an edge of G with maximum length, let $\mathcal{T}$ be the SPQ-tree of G such that the Q-node corresponding to e is a child of the root of $\mathcal{T}$, and let h be the height of $\mathcal{T}$. There exists an $n^{O(2^h)}$-time algorithm that tests whether G admits a planar straight-line realization and, in the positive case, constructs such a realization.*

Due to Lemma 20, Theorem 15 implies the following.

Corollary 1. *Let G an n-vertex weighted 2-tree and let ℓ be length of the longest path of G. There exists an $n^{O(4^\ell)}$-time algorithm that tests whether G admits a planar straight-line realization and, in the positive case, constructs such a realization.*

In the remainder of the section, we prove Theorem 15. Let H be a 2-tree and let (u, v) be an edge of H. A *uv-external* realization of H is a planar straight-line realization of H such that both u and v are incident to the outer face; note that the edge (u, v) is not necessarily incident to the outer face of a uv-external realization of H. Given two uv-external realizations Γ and Γ' of H, we say that Γ is *uv-equivalent* to Γ' if Γ can be rotated and translated (note that reflections are not allowed) so that the representation of (u, v) coincides with the one in Γ' and so that the boundary of the outer face of Γ coincides with the boundary of the outer face of Γ'. Given a set $\mathcal{R}$ of uv-external realizations of H, we say that $\mathcal{R}$ is *uv-complete* if, for every uv-external realization Γ' of H, there exists a uv-external realization Γ of H in $\mathcal{R}$ such that Γ is uv-equivalent to Γ'. Finally, a uv-complete set $\mathcal{R}$ of uv-external realizations of H is *minimal* if no realization in $\mathcal{R}$ is uv-equivalent to a distinct realization in $\mathcal{R}$.

As a first step of our algorithm, we compute the SPQ-tree $\mathcal{T}$ of G such that the Q-node corresponding to e is a child of the root ρ of $\mathcal{T}$; this can be done in $O(n)$ time [9,33,53]. One key point of our algorithm is that, for every P-node μ of $\mathcal{T}$, we only need to look for $u_\mu v_\mu$-external realizations of G_μ (recall that u_μ and v_μ are the poles of μ). This is formalized in the following.

Lemma 21. *Let Γ be any planar straight-line realization of G. For any P-node μ of $\mathcal{T}$, the restriction Γ_μ of Γ to G_μ is a $u_\mu v_\mu$-external realization of G_μ.*

Proof. Consider first the case in which $\mu = \rho$. Then Γ_μ coincides with Γ. If Γ were not a $u_\mu v_\mu$-external realization of G_μ, then e would lie inside a 3-cycle c of G in Γ. However, this would contradict the planarity of Γ. Namely, since e is an edge with maximum length, the length of e would be larger than the one of each side of the triangle representing c in Γ.

Consider now the case in which $\mu \neq \rho$. Suppose, for a contradiction, that (at least) one of u_μ and v_μ, say u_μ, is not incident to the outer face of Γ_μ. Hence, u_μ lies inside a cycle c of G_μ in Γ_μ and in Γ. Furthermore, at most one of u_ρ and v_ρ, say u_ρ, is a vertex of c (if both u_ρ and v_ρ would belong to G_μ, we would have $\mu = \rho$). Since u_ρ and v_ρ are incident to the outer face of Γ, we have that v_ρ lies outside c in Γ. Since G is biconnected, it contains a path that connects u_μ with v_ρ and that does not contain vertices of G_μ other than u_μ. By the Jordan curve theorem, such a path crosses c in Γ, a contradiction. □

For each P-node μ of $\mathcal{T}$, we construct an order $\nu_1, \ldots, \nu_k$ of the children of μ as follows. First, we let ν_1 be the only Q-node that is a child of μ; this node exists by Condition (C2). We denote by $\lambda(\nu_1)$ the length of the edge of G corresponding to such a Q-node. Every other child ν_i of μ is an S-node. Let σ_i and τ_i be the two children of ν_i; by Condition (C3), each of them is either a Q-node or a P-node. We denote by w_{ν_i} the pole that σ_i and τ_i share (that is, their pole different from u_μ and v_μ); also, we can assume, w.l.o.g. up to a relabeling of σ_i with τ_i, that u_μ is a pole of σ_i and v_μ is a pole of τ_i. Then the edges (u_μ, w_{ν_i}) and (v_μ, w_{ν_i}) belong to G_{ν_i}, as each of σ_i and τ_i is either a Q-node (and then it obviously contains the edge between its poles) or a P-node (and then it contains the edge between its poles by Condition (C2)). Let $\lambda(\nu_i)$ denote the sum of the lengths of the edges (u_μ, w_{ν_i}) and (v_μ, w_{ν_i}). Then the order of the children of μ is established so that $\lambda(\nu_1) \leq \lambda(\nu_2) \leq \cdots \leq \lambda(\nu_k)$. Clearly, such orders can be found for all the P-nodes of $\mathcal{T}$ in overall $O(n \log n)$ time. For a P-node μ of $\mathcal{T}$ with children $\nu_1, \ldots, \nu_k$ and for any $i \in \{1, \ldots, k\}$, we denote by G^i_μ the subgraph of G_μ which

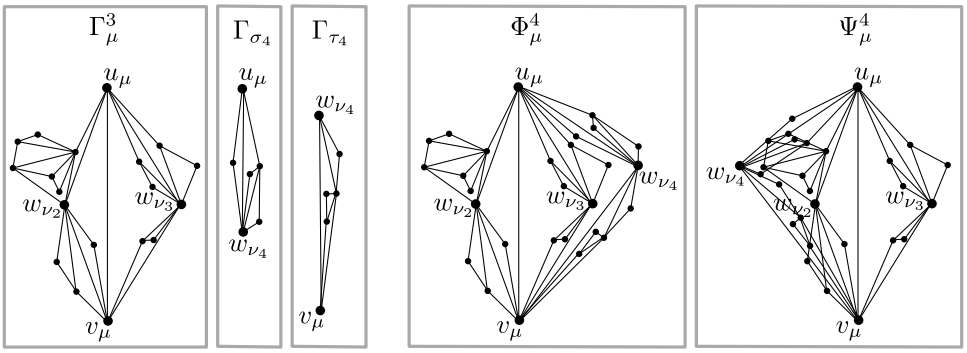

Fig. 31. Construction of the realizations Φ^i_μ and Ψ^i_μ of G^i_μ from the realizations $\Gamma^{i-1}_\mu \in \mathcal{R}^{i-1}_\mu$, $\Gamma_{\sigma_i} \in \mathcal{R}_{\sigma_i}$, and $\Gamma_{\tau_i} \in \mathcal{R}_{\tau_i}$. In this example, $i = 4$; further, Φ^i_μ is planar and Ψ^i_μ is non-planar.

is the union of the graphs $G_{\nu_1}, \ldots, G_{\nu_i}$. Observe that G^1_μ is the edge (u_μ, v_μ), while G^k_μ coincides with G_μ.

Our algorithm either concludes that G admits no planar straight-line realization or constructs the following sets:

- For each Q-node or P-node μ, the algorithm constructs a non-empty minimal $u_\mu v_\mu$-complete set $\mathcal{R}_\mu$ of $u_\mu v_\mu$-external realizations of G_μ. For each realization Γ_μ in $\mathcal{R}_\mu$, the algorithm also constructs the plane embedding $\mathcal{E}(\Gamma_\mu)$.
- For each P-node μ with children $\nu_1, \ldots, \nu_k$ and for each $i = 1, \ldots, k$, the algorithm constructs a non-empty minimal $u_\mu v_\mu$-complete set $\mathcal{R}^i_\mu$ of $u_\mu v_\mu$-external realizations of G^i_μ. For each realization Γ^i_μ in $\mathcal{R}^i_\mu$, the algorithm also constructs the plane embedding $\mathcal{E}(\Gamma^i_\mu)$.

In order to construct such sets, the algorithm bottom-up visits all the Q-nodes and all the P-nodes of $\mathcal{T}$.

When a Q-node μ is visited, the set $\mathcal{R}_\mu$ is defined as $\{\Gamma_\mu\}$, where Γ_μ is the unique (up to rotation and translation) straight-line realization of the edge of G corresponding to μ. Obviously, $\mathcal{R}_\mu$ is a minimal $u_\mu v_\mu$-complete set of $u_\mu v_\mu$-external realizations of G_μ.

Suppose now that a P-node μ of $\mathcal{T}$ with children $\nu_1, \ldots, \nu_k$ is considered. For $i = 2, \ldots, k$, since we already visited the children σ_i and τ_i of ν_i, we have the sets $\mathcal{R}_{\sigma_i}$ and $\mathcal{R}_{\tau_i}$ (indeed, if we concluded that G admits no planar straight-line realization, there is nothing else to do). Our algorithm processes the children $\nu_1, \ldots, \nu_k$ of μ in this order. For $i = 1, \ldots, k$, after visiting ν_i, the algorithm either concludes that G admits no planar straight-line realization or constructs the set $\mathcal{R}^i_\mu$. In the latter case, for each realization Γ^i_μ in $\mathcal{R}^i_\mu$, the algorithm also constructs the plane embedding $\mathcal{E}(\Gamma^i_\mu)$. If the algorithm constructs the set $\mathcal{R}^k_\mu$, then we set $\mathcal{R}_\mu = \mathcal{R}^k_\mu$.

We now describe how the algorithm processes the children $\nu_1, \ldots, \nu_k$ of a P-node μ. The algorithm starts by defining $\mathcal{R}^1_\mu = \mathcal{R}_{\nu_1}$ (recall that ν_1 is a Q-node). Then, for $i = 2, \ldots, k$, it constructs the set $\mathcal{R}^i_\mu$ by combining the realizations of G^{i-1}_μ in $\mathcal{R}^{i-1}_\mu$ with the realizations of G_{σ_i} in $\mathcal{R}_{\sigma_i}$ and with the realizations of G_{τ_i} in $\mathcal{R}_{\tau_i}$. This is done as follows.

We initialize $\mathcal{R}^i_\mu = \emptyset$. Then, for every triple of realizations $\Gamma^{i-1}_\mu \in \mathcal{R}^{i-1}_\mu$, $\Gamma_{\sigma_i} \in \mathcal{R}_{\sigma_i}$, and $\Gamma_{\tau_i} \in \mathcal{R}_{\tau_i}$, we construct two straight-line realizations Φ^i_μ and Ψ^i_μ of G^i_μ (and two plane embeddings $\mathcal{E}(\Phi^i_\mu)$ and $\mathcal{E}(\Psi^i_\mu)$ of G^i_μ) as follows. Recall that the poles of σ_i are u_μ and w_{ν_i}, while the ones of τ_i are v_μ and w_{ν_i}. Refer to Figs. 31 and 32.

- First, we initialize both Φ^i_μ and Ψ^i_μ to Γ^{i-1}_μ.
- Second, we place the vertex w_{ν_i} in Φ^i_μ and Ψ^i_μ with the two placements that realize the prescribed lengths of the edges (u_μ, w_{ν_i}) and (v_μ, w_{ν_i}); in particular, the placement of w_{ν_i} in

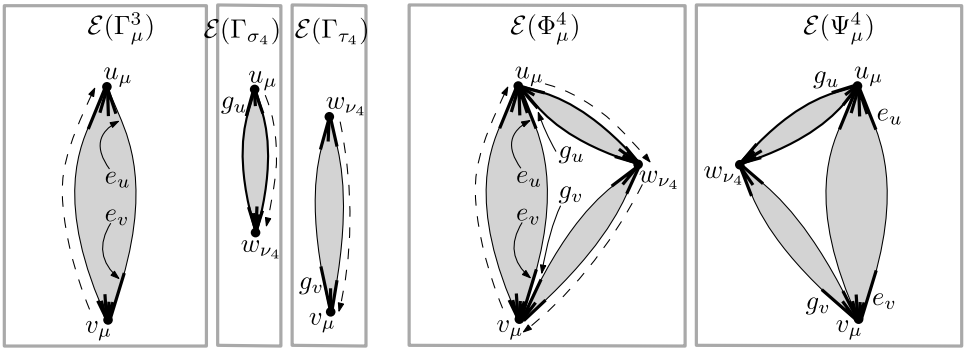

Fig. 32. Construction of the plane embeddings $\mathcal{E}(\Phi_\mu^i)$ and $\mathcal{E}(\Psi_\mu^i)$ of G_μ^i from the plane embeddings $\mathcal{E}(\Gamma_\mu^{i-1})$, $\mathcal{E}(\Gamma_{\sigma_i})$, and $\mathcal{E}(\Gamma_{\tau_i})$. The illustration is coherent with the realizations of Fig. 31.

Φ_μ^i is such that the clockwise order of the vertices of the cycle $(u_\mu, v_\mu, w_{\nu_i})$ is u_μ, w_{ν_i}, v_μ, while the placement of w_{ν_i} in Ψ_μ^i is such that the clockwise order of the vertices of the cycle $(u_\mu, v_\mu, w_{\nu_i})$ is u_μ, v_μ, w_{ν_i}.

– Third, we insert Γ_{σ_i} in Φ_μ^i so that the representation of the edge (u_μ, w_{ν_i}) in Γ_{σ_i} coincides with the one in Φ_μ^i. We also insert Γ_{τ_i} in Φ_μ^i so that the representation of the edge (v_μ, w_{ν_i}) in Γ_{τ_i} coincides with the one in Φ_μ^i. This concludes the construction of Φ_μ^i. The insertion of Γ_{σ_i} and Γ_{τ_i} in Ψ_μ^i in order to conclude the construction of Ψ_μ^i is done in the same way.
– Finally, we construct a plane embedding $\mathcal{E}(\Phi_\mu^i)$ as follows:

  - We glue the lists of the edges incident to u_μ in $\mathcal{E}(\Gamma_\mu^{i-1})$ and $\mathcal{E}(\Gamma_{\sigma_i})$, so that, in the counter-clockwise order of the edges incident to u_μ, the edge e_u that enters u_μ when traversing in counter-clockwise order the boundary of the outer face of $\mathcal{E}(\Gamma_\mu^{i-1})$ comes immediately before the edge g_u that enters u_μ when traversing in clockwise order the boundary of the outer face of $\mathcal{E}(\Gamma_{\sigma_i})$.
  - We glue the lists of the edges incident to v_μ in $\mathcal{E}(\Gamma_\mu^{i-1})$ and $\mathcal{E}(\Gamma_{\tau_i})$, so that, in the clockwise order of the edges incident to v_μ, the edge e_v that enters v_μ when traversing in clockwise order the boundary of the outer face of Γ_μ^{i-1} comes immediately before the edge g_v that enters v_μ when traversing in counter-clockwise order the boundary of the outer face of Γ_{τ_i}.
  - We let the cycle bounding the outer face of $\mathcal{E}(\Phi_\mu^i)$ be composed of the path that goes from v_μ to u_μ in clockwise direction along the outer face of $\mathcal{E}(\Gamma_\mu^{i-1})$, of the path that goes from u_μ to w_{ν_i} in clockwise direction along the outer face of $\mathcal{E}(\Gamma_{\sigma_i})$, and of the path that goes from w_{ν_i} to v_μ in clockwise direction along the outer face of $\mathcal{E}(\Gamma_{\tau_i})$.

  We remark that $\mathcal{E}(\Phi_\mu^i)$ is not necessarily the plane embedding of Φ_μ^i, in fact Φ_μ^i might even be non-planar.
  A plane embedding $\mathcal{E}(\Psi_\mu^i)$ is constructed analogously.

The straight-line realizations Φ_μ^i and Ψ_μ^i, together with the plane embeddings $\mathcal{E}(\Phi_\mu^i)$ and $\mathcal{E}(\Psi_\mu^i)$, are hence constructed in $O(n)$ time.

We remark that the plane embeddings $\mathcal{E}(\Phi_\mu^i)$ and $\mathcal{E}(\Psi_\mu^i)$ both place G_{ν_i} "externally" with respect to G_μ^{i-1}. This is a main ingredient for bounding the running time of our algorithm. It is also not a loss of generality, because of the ordering that has been established for the children of μ, as proved in the following lemma.

Lemma 22. *Consider a $u_\mu v_\mu$-external realization Γ_μ^i of G_μ^i. Then the restriction of Γ_μ^i to G_{v_i} lies in the outer face of the restriction of Γ_μ^i to G_μ^{i-1}.*

Proof. Suppose, for a contradiction, that the restriction of Γ_μ^i to G_{v_i} lies inside an internal face f of the restriction of Γ_μ^i to G_μ^{i-1}. In particular, the path (u_μ, w_{v_i}, v_μ) lies inside f in Γ_μ^i. Since Γ_μ^i is a $u_\mu v_\mu$-external realization of G_μ^i, both u_μ and v_μ are incident to the outer face of Γ_μ^i, hence it is possible to draw a Jordan arc $\mathcal{C}_\mu$ connecting u_μ and v_μ and lying in the outer face of Γ_μ^i, except at its end-points. Now $\mathcal{C}_\mu$ and the path (u_μ, w_{v_i}, v_μ) form a closed curve which contains the edge (u_μ, v_μ) on one side and a graph G_{v_j} on the other side (as otherwise f would be the outer face of Γ_μ^i), for some $2 \leq j < i$. However, this implies that the path (u_μ, w_{v_i}, v_μ) lies inside the triangle representing the 3-cycle (u_μ, w_{v_j}, v_μ) in Γ_μ^i. This contradicts the planarity of Γ_μ^i; indeed, we have $\ell(v_i) > \ell(v_j) > \ell(v_1)$, hence the length of the path (u_μ, w_{v_i}, v_μ) is larger than the lengths of the path (u_μ, w_{v_j}, v_μ) and of the edge (u_μ, v_μ) in Γ_μ^i. □

We now resume the description of how the algorithm constructs $\mathcal{R}_\mu^i$. By means of Theorem 4, we test in $O(n)$ time whether Φ_μ^i is a planar straight-line realization respecting $\mathcal{E}(\Phi_\mu^i)$. By the construction of $\mathcal{E}(\Phi_\mu^i)$, the test is positive if and only if Φ_μ^i is a $u_\mu v_\mu$-external realization of G_μ^i. If the test is negative, then we discard Φ_μ^i. Otherwise, for every realization Λ_μ^i in $\mathcal{R}_\mu^i$, we test whether Φ_μ^i is $u_\mu v_\mu$-equivalent to Λ_μ^i and, if it is, we discard Φ_μ^i. This can be done in $O(n)$ time by rotating and translating Φ_μ^i so that the representation of the edge (u_μ, v_μ) coincides with the one in Λ_μ^i and by then checking whether the vertices along on the boundary of the outer face of Φ_μ^i have the same coordinates as the vertices along on the boundary of the outer face of Λ_μ^i. If we did not discard Φ_μ^i, then we insert it into $\mathcal{R}_\mu^i$.

We process Ψ_μ^i analogously to Φ_μ^i.

After we have processed every triple of drawings $\Gamma_\mu^{i-1} \in \mathcal{R}_\mu^{i-1}$, $\Gamma_{\sigma_i} \in \mathcal{R}_{\sigma_i}$, and $\Gamma_{\tau_i} \in \mathcal{R}_{\tau_i}$, if the set $\mathcal{R}_\mu^i$ is empty, then we report that G admits no planar straight-line realization, otherwise $\mathcal{R}_\mu^i$ is a minimal $u_\mu v_\mu$-complete set of $u_\mu v_\mu$-external realizations of G_μ^i. The correctness of these conclusions is proved in the following lemma.

Lemma 23. *If $\mathcal{R}_\mu^i$ is empty, then G admits no planar straight-line realization. Otherwise, $\mathcal{R}_\mu^i$ is a minimal $u_\mu v_\mu$-complete set of $u_\mu v_\mu$-external realizations of G_μ^i.*

Proof. The main ingredient of the proof is the following claim: If G_μ^i admits a $u_\mu v_\mu$-external realization Λ_μ^i, then $\mathcal{R}_\mu^i$ contains a $u_\mu v_\mu$-external realization that is $u_\mu v_\mu$-equivalent to Λ_μ^i.

We prove the claim. For a contradiction, suppose that G_μ^i admits a $u_\mu v_\mu$-external realization Λ_μ^i and that $\mathcal{R}_\mu^i$ does not contain any $u_\mu v_\mu$-external realization that is $u_\mu v_\mu$-equivalent to Λ_μ^i. We are going to replace parts of Λ_μ^i with realizations in $\mathcal{R}_\mu^{i-1}$, $\mathcal{R}_{\sigma_i}$, and $\mathcal{R}_{\tau_i}$, so to obtain a $u_\mu v_\mu$-external realization of G_μ^i that is constructed by the algorithm and that should hence have been inserted in $\mathcal{R}_\mu^i$ (unless a $u_\mu v_\mu$-equivalent $u_\mu v_\mu$-external realization of G_μ^i is already in $\mathcal{R}_\mu^i$), from which a contradiction follows.

Let Λ_μ^{i-1}, Λ_{σ_i}, and Λ_{τ_i} be the restrictions of Λ_μ^i to G_μ^{i-1}, G_{σ_i}, and G_{τ_i}, respectively. By Lemma 22, we have that Λ_{σ_i} and Λ_{τ_i} lie in the outer face of Λ_μ^{i-1}; furthermore, the fact that Λ_μ^i is a $u_\mu v_\mu$-external realization directly implies that Λ_{σ_i} and Λ_μ^{i-1} lie in the outer face of Λ_{τ_i} and that Λ_{τ_i} and Λ_μ^{i-1} lie in the outer face of Λ_{σ_i}.

Since $\mathcal{R}_\mu^{i-1}$ is a $u_\mu v_\mu$-complete set of $u_\mu v_\mu$-external realizations of G_μ^{i-1}, it follows that there exists a $u_\mu v_\mu$-external realization $\Gamma_\mu^{i-1} \in \mathcal{R}_\mu^{i-1}$ of G_μ^{i-1} that is $u_\mu v_\mu$-equivalent to Λ_μ^{i-1}. Analogously, there exists a $u_\mu w_{v_i}$-external realization $\Gamma_{\sigma_i} \in \mathcal{R}_{\sigma_i}$ of G_{σ_i} that is $u_\mu w_{v_i}$-equivalent to Λ_{σ_i} and there exists a $v_\mu w_{v_i}$-external realization $\Gamma_{\tau_i} \in \mathcal{R}_{\tau_i}$ of G_{τ_i} that is $v_\mu w_{v_i}$-equivalent to Λ_{τ_i}.

We can now replace Λ_μ^{i-1} with Γ_μ^{i-1} in Λ_μ^i, after rotating and translating Γ_μ^{i-1} so that the representation of the edge (u_μ, v_μ) coincides with the one in Λ_μ^i. Since the boundary of the outer face of Γ_μ^{i-1} coincides with the boundary of the outer face of Λ_μ^{i-1} and since Λ_{σ_i} and Λ_{τ_i} lie in the

outer face of Λ_μ^{i-1}, it follows that after the replacement Λ_μ^i remains a $u_\mu v_\mu$-external realization of G_μ^i (in particular, it remains planar) that is $u_\mu v_\mu$-equivalent to Λ_μ^i. We can analogously replace Λ_{σ_i} with Γ_{σ_i} and Λ_{τ_i} with Γ_{τ_i}, obtaining a $u_\mu v_\mu$-external realization of G_μ that is constructed by the algorithm and that is $u_\mu v_\mu$-equivalent to Λ_μ^i. This implies that $\mathcal{R}_\mu^i$ contains a $u_\mu v_\mu$-external realization that is $u_\mu v_\mu$-equivalent to Λ_μ^i, a contradiction which proves the claim.

The claim directly implies that, if $\mathcal{R}_\mu^i$ is empty, then G_μ^i admits no $u_\mu v_\mu$-external realization, hence G_μ admits no $u_\mu v_\mu$-external realization (given that G_μ^i is a subgraph of G_μ), and finally that G admits no planar straight-line realization, by Lemma 21.

Suppose now that $\mathcal{R}_\mu^i$ is non-empty. We prove that $\mathcal{R}_\mu^i$ is a minimal $u_\mu v_\mu$-complete set of $u_\mu v_\mu$-external realizations of G_μ^i. First, that every realization Γ_μ^i in $\mathcal{R}_\mu^i$ is $u_\mu v_\mu$-external directly descends from the $O(n)$-time test that is performed by means of Theorem 4 on Γ_μ^i; indeed, Γ_μ^i is inserted into $\mathcal{R}_\mu^i$ only if the test reports that Γ_μ^i is a planar straight-line realization whose plane embedding is the one constructed by the algorithm, in which u_μ and v_μ are incident to the outer face. Second, that $\mathcal{R}_\mu^i$ is $u_\mu v_\mu$-complete is a direct consequence of the claim above. Finally, the minimality of $\mathcal{R}_\mu^i$ comes from the fact that a realization is inserted into $\mathcal{R}_\mu^i$ only if it is not $u_\mu v_\mu$-equivalent to any realization already in $\mathcal{R}_\mu^i$. □

It remains to determine the running time of our algorithm. The main ingredient we are going to use is the following combinatorial lemma.

Lemma 24. *Let μ be a Q-node or a P-node of $\mathcal{T}$, and let η be the height of $\mathcal{T}_\mu$. We have that $|\mathcal{R}_\mu| \leq n^{\frac{2^\eta-1}{3}} \cdot 2^{\frac{2^\eta-1}{3}}$. Furthermore, if μ is a P-node, let $\nu_1, \ldots, \nu_k$ be the children of μ. For $i = 1, \ldots, k$, we have that $|\mathcal{R}_\mu^i| \leq n^{\frac{2^\eta-1}{3}} \cdot 2^{\frac{2^\eta-1}{3}}$.*

Proof. We prove the following claim. Let $f(k)$ be the function whose domain consists of the even natural numbers, that is recursively defined as follows: $f(k) = 1$ if $k = 0$, and $f(k) = 2n \cdot (f(k-2))^4$ if $k > 0$. Then we claim that $|\mathcal{R}_\mu| \leq f(\eta)$ and $|\mathcal{R}_\mu^i| \leq f(\eta)$. The claim implies the lemma, as an easy inductive proof shows that the closed form of $f(k)$ is $f(k) = n^{\frac{2^k-1}{3}} \cdot 2^{\frac{2^k-1}{3}}$.

The claim is proved by induction on η. If $\eta = 0$, then μ is a Q-node, hence by construction we have that $|\mathcal{R}_\mu| = 1 = f(0)$.

Suppose now that $\eta > 0$, hence μ is a P-node with children $\nu_1, \ldots, \nu_k$. First, we have $\mathcal{R}_\mu^1 = 1 < f(\eta)$. We now prove that $|\mathcal{R}_\mu^i| \leq n^{\frac{2^\eta-1}{3}} \cdot 2^{\frac{2^\eta-1}{3}}$, for any $i \in \{2, \ldots, k\}$; this suffices to complete the induction, since by construction $\mathcal{R}_\mu = \mathcal{R}_\mu^k$. Consider any realization Γ_μ^i in $\mathcal{R}_\mu^i$. Since Γ_μ^i is a $u_\mu v_\mu$-external realization of G_μ^i and since $i \geq 2$, it follows that there are exactly two graphs among $G_{\nu_1}, \ldots, G_{\nu_i}$ that contain edges incident to the outer face of Γ_μ^i. By Lemma 22, one of these graphs is G_{ν_i}, while the other graph, say G_{ν_j}, might be any graph among $G_{\nu_1}, \ldots, G_{\nu_{i-1}}$. When traversing the boundary of the outer face of Γ_μ^i in clockwise direction, the edge e_{ν_i} of G_{ν_i} incident to u_μ might appear right before or right after u_μ. Altogether, there are less than $2n$ possibilities for the choice of the graph G_{ν_j} and for whether e_{ν_i} appears right before or right after u_μ in clockwise order along the boundary of the outer face of Γ_μ^i. For each of these $2n$ possibilities, the interior of Γ_μ^i is determined by the choice of four realizations $\Gamma_{\sigma_i} \in \mathcal{R}_{\sigma_i}$, $\Gamma_{\tau_i} \in \mathcal{R}_{\tau_i}$, $\Gamma_{\sigma_j} \in \mathcal{R}_{\sigma_j}$, and $\Gamma_{\tau_j} \in \mathcal{R}_{\tau_j}$. Indeed, the restriction of Γ_μ^i to G_{σ_i}, G_{τ_i}, G_{σ_j}, and G_{τ_j} is a realization in $\mathcal{R}_{\sigma_i}$, $\mathcal{R}_{\tau_i}$, $\mathcal{R}_{\sigma_j}$, and $\mathcal{R}_{\tau_j}$, respectively. Since the heights of $\mathcal{T}_{\sigma_i}$, $\mathcal{T}_{\tau_i}$, $\mathcal{T}_{\sigma_j}$, and $\mathcal{T}_{\tau_j}$ are each at most $\eta - 2$, by induction each of $\mathcal{R}_{\sigma_i}$, $\mathcal{R}_{\tau_i}$, $\mathcal{R}_{\sigma_j}$, and $\mathcal{R}_{\tau_j}$ has size smaller than or equal to $f(\eta-2)$. Hence, the number of possible choices for the realizations $\Gamma_{\sigma_i} \in \mathcal{R}_{\sigma_i}$, $\Gamma_{\tau_i} \in \mathcal{R}_{\tau_i}$, $\Gamma_{\sigma_j} \in \mathcal{R}_{\sigma_j}$, and $\Gamma_{\tau_j} \in \mathcal{R}_{\tau_j}$ is smaller than or equal to $(f(\eta-2))^4$. Finally, the minimality of $\mathcal{R}_\mu^i$ implies that no two realizations in $\mathcal{R}_\mu^i$ coincide when restricted to G_{σ_i}, G_{τ_i}, G_{σ_j}, and G_{τ_j}. This proves that the number of realizations in $\mathcal{R}_\mu^i$ is smaller than or equal to $2n \cdot (f(\eta-2))^4$; this concludes the induction and hence the proof of the lemma. □

We are now ready to determine the running time of our algorithm. Whenever a P-node μ with k children $\nu_1, \ldots, \nu_k$ is visited in the bottom-up traversal of $\mathcal{T}$, for $i = 1, \ldots, k$, the set $\mathcal{R}_\mu^i$ is

constructed by considering every triple of drawings $\Gamma_\mu^{i-1} \in \mathcal{R}_\mu^{i-1}$, $\Gamma_{\sigma_i} \in \mathcal{R}_{\sigma_i}$, and $\Gamma_{\tau_i} \in \mathcal{R}_{\tau_i}$, where σ_i and τ_i are the children of ν_i. By Lemma 24, the number of such triples is at most $\left(n^{\frac{2^h-1}{3}} \cdot 2^{\frac{2^h-1}{3}}\right)\left(n^{\frac{2^{(h-2)}-1}{3}} \cdot 2^{\frac{2^{(h-2)}-1}{3}}\right)^2 = n^{(2^{(h-1)}-1)} \cdot 2^{(2^{(h-1)}-1)}$; indeed, the height of $\mathcal{T}$ is h, hence the height of $\mathcal{T}_\mu$ is smaller than or equal to h, and the heights of $\mathcal{T}_{\sigma_i}$ and $\mathcal{T}_{\tau_i}$ are smaller than or equal to $h-2$. For each of such triples, two straight-line realizations Φ_μ^i and Ψ_μ^i are constructed in $O(n)$ time, together with the plane embeddings $\mathcal{E}(\Phi_\mu^i)$ and $\mathcal{E}(\Psi_\mu^i)$. By means of Theorem 4, it is tested in $O(n)$ time whether Φ_μ^i and Ψ_μ^i are planar straight-line realizations with plane embeddings $\mathcal{E}(\Phi_\mu^i)$ and $\mathcal{E}(\Psi_\mu^i)$, respectively. For each realization Λ_μ^i of G_μ^i in $\mathcal{R}_\mu^i$, it is tested in $O(n)$ time whether Φ_μ^i is $u_\mu v_\mu$-equivalent to Λ_μ^i. By Lemma 24, the set $\mathcal{R}_\mu^i$ contains at most $\left(n^{\frac{2^h-1}{3}} \cdot 2^{\frac{2^h-1}{3}}\right)$ realizations of G_μ^i, hence these $O(n)$-time tests are performed $\left(n^{(2^{(h-1)}-1)} \cdot 2^{(2^{(h-1)}-1)}\right)\left(n^{\frac{2^h-1}{3}} \cdot 2^{\frac{2^h-1}{3}}\right) = \left(n^{\frac{5\cdot 2^{h-1}-4}{3}} \cdot 2^{\frac{5\cdot 2^{h-1}-4}{3}}\right)$ times, with a $\left(n^{\frac{5\cdot 2^{h-1}-1}{3}} \cdot 2^{\frac{5\cdot 2^{h-1}-4}{3}}\right)$ running time. Since the number of P-nodes in $\mathcal{T}$ is in $O(n)$, the running time of our algorithm is in $\left(n^{\frac{5\cdot 2^{h-1}+2}{3}} \cdot 2^{\frac{5\cdot 2^{h-1}-4}{3}}\right) \in n^{O(2^h)}$. This concludes the proof of Theorem 15.

8. Conclusions and open problems

We studied the problem of testing if a 2-tree has a planar straight-line drawing with prescribed edge lengths, both when the number of distinct edge lengths is bounded and when it is unbounded.

In the bounded setting, we proved that the problem is linear-time solvable if the number of prescribed distinct lengths is at most two, while it is NP-hard if this number is four or more. Hence, an intriguing question is to determine the complexity of the problem for weighted 2-trees with three prescribed distinct lengths.

In the unbounded setting, we showed that the problem is linear-time solvable if the embedding of the input graph is prescribed. Moreover, we studied weighted maximal outerplanar graphs, and we proved that the problem is linear-time solvable if their dual tree is a path, and cubic-time solvable if their dual tree is a caterpillar. We leave open the problem about general maximal outerplanar graphs.

Finally, for the general problem on weighted 2-trees, we described a slice-wise polynomial-time algorithm, parameterized by the length of the longest path. A natural open problem is to determine whether the problem admits an FPT algorithm with respect to the same parameter.

References

[1] Z. Abel, E.D. Demaine, M.L. Demaine, S. Eisenstat, J. Lynch, T.B. Schardl, Who needs crossings? Hardness of plane graph rigidity, in: S.P. Fekete, A. Lubiw (Eds.), 32nd International Symposium on Computational Geometry, SoCG '16, in: LIPIcs, vol. 51, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016, pp. 3:1–3:15, http://dx.doi.org/10.4230/LIPIcs.SoCG.2016.3.

[2] O. Aichholzer, V. Kusters, W. Mulzer, A. Pilz, M. Wettstein, An optimal algorithm for reconstructing point set order types from radial orderings, Internat. J. Comput. Geom. Appl. 27 (1–2) (2017) 57–84.

[3] N. Alon, O.N. Feldheim, Drawing outerplanar graphs using three edge lengths, Comput. Geom. 48 (3) (2015) 260–267, http://dx.doi.org/10.1016/j.comgeo.2014.10.006.

[4] P. Angelini, G. Da Lozzo, G. Di Battista, V. Di Donato, P. Kindermann, G. Rote, I. Rutter, Windrose planarity: Embedding graphs with direction-constrained edges, ACM Trans. Algorithms 14 (4) (2018) 54:1–54:24, http://dx.doi.org/10.1145/3239561.

[5] P. Angelini, G.D. Lozzo, M.D. Bartolomeo, G.D. Battista, S. Hong, M. Patrignani, V. Roselli, Anchored drawings of planar graphs, in: C.A. Duncan, A. Symvonis (Eds.), 22nd International Symposium on Graph Drawing, GD '14, in: Lecture Notes in Computer Science, vol. 8871, Springer, 2014, pp. 404–415, http://dx.doi.org/10.1007/978-3-662-45803-7_34.

[6] B. Aspvall, M.F. Plass, R.E. Tarjan, A linear-time algorithm for testing the truth of certain quantified boolean formulas, Inform. Process. Lett. 8 (3) (1979) 121–123, http://dx.doi.org/10.1016/0020-0190(79)90002-4.

[7] Z. Bakhajian, O.N. Feldheim, Drawing outerplanar graphs using thirteen edge lengths, Comput. Geom. 110 (2023) 101964, http://dx.doi.org/10.1016/j.comgeo.2022.101964.

[8] G.D. Battista, P. Eades, R. Tamassia, I.G. Tollis, Graph Drawing: Algorithms for the Visualization of Graphs, Prentice-Hall, 1999.

[9] G.D. Battista, R. Tamassia, On-line planarity testing, SIAM J. Comput. 25 (5) (1996) 956–997, http://dx.doi.org/10.1137/S0097539794280736.

[10] G.D. Battista, R. Tamassia, I.G. Tollis, Constrained visibility representations of graphs, Inf. Process. Lett. 41 (1) (1992) 1–7, http://dx.doi.org/10.1016/0020-0190(92)90072-4.
[11] M. Ben-Or, Lower bounds for algebraic computation trees (preliminary report), in: D.S. Johnson, R. Fagin, M.L. Fredman, D. Harel, R.M. Karp, N.A. Lynch, C.H. Papadimitriou, R.L. Rivest, W.L. Ruzzo, J.I. Seiferas (Eds.), 15th Annual ACM Symposium on Theory of Computing, STOC '83, ACM, 1983, pp. 80–86, http://dx.doi.org/10.1145/800061.808735.
[12] M. de Berg, A. Khosravi, Optimal binary space partitions for segments in the plane, Internat. J. Comput. Geom. Appl. 22 (3) (2012) 187–206, http://dx.doi.org/10.1142/S0218195912500045.
[13] B. Berger, J.M. Kleinberg, F.T. Leighton, Reconstructing a three-dimensional model with arbitrary errors, J. ACM 46 (2) (1999) 212–235, http://dx.doi.org/10.1145/301970.301972.
[14] V. Blazej, J. Fiala, G. Liotta, On the edge-length ratio of 2-trees, in: D. Auber, P. Valtr (Eds.), 28th International Symposium on Graph Drawing and Network Visualization, GD '20, in: Lecture Notes in Computer Science, vol. 12590, Springer, 2020, pp. 85–98, http://dx.doi.org/10.1007/978-3-030-68766-3_7.
[15] M. Borrazzo, F. Frati, On the planar edge-length ratio of planar graphs, J. Comput. Geom. 11 (1) (2020) 137–155, http://dx.doi.org/10.20382/jocg.v11i1a6.
[16] U. Brandes, B. Schlieper, Angle and distance constraints on tree drawings, in: M. Kaufmann, D. Wagner (Eds.), 14th International Symposium on Graph Drawing, GD '06, in: Lecture Notes in Computer Science, vol. 4372, Springer, 2006, pp. 54–65, http://dx.doi.org/10.1007/978-3-540-70904-6_7.
[17] A. Brandstädt, V.B. Le, J.P. Spinrad, Graph classes: A survey, in: SIAM Monographs on Discrete Mathematics and Applications, SIAM, 1999.
[18] S. Cabello, E.D. Demaine, G. Rote, Planar embeddings of graphs with specified edge lengths, J. Graph Algorithms Appl. 11 (1) (2007) 259–276, http://dx.doi.org/10.7155/jgaa.00145.
[19] S. Capkun, M. Hamdi, J. Hubaux, GPS-free positioning in mobile ad hoc networks, Clust. Comput. 5 (2) (2002) 157–167, http://dx.doi.org/10.1023/A:1013933626682.
[20] P. Carmi, V. Dujmovic, P. Morin, D.R. Wood, Distinct distances in graph drawings, Electron. J. Combin. 15 (1) (2008) http://www.combinatorics.org/Volume_15/Abstracts/v15i1r107.html.
[21] S. Chaplick, P. Kindermann, A. Löffler, F. Thiele, A. Wolff, A. Zaft, J. Zink, Recognizing stick graphs with and without length constraints, J. Graphs Algorithms Appl. 24 (4) (2020) 657–681, http://dx.doi.org/10.7155/jgaa.00524.
[22] B. Chazelle, Triangulating a simple polygon in linear time, Discrete Comput. Geom. 6 (1991) 485–524, http://dx.doi.org/10.1007/BF02574703.
[23] R. Connelly, On generic global rigidity, in: P. Gritzmann, B. Sturmfels (Eds.), Proceedings of a DIMACS Workshop on Applied Geometry and Discrete Mathematics, in: DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 4, DIMACS/AMS, 1990, pp. 147–156, http://dx.doi.org/10.1090/dimacs/004/11.
[24] C.R. Coullard, A. Lubiw, Distance visibility graphs, Internat. J. Comput. Geom. Appl. 2 (4) (1992) 349–362, http://dx.doi.org/10.1142/S0218195992000202.
[25] T. Deng, On The Implementation and Refinement of Outerplanar Graph Algorithms, (Master's thesis), University of Windsor, Ontario, Canada, 2007.
[26] O. Devillers, G. Liotta, F.P. Preparata, R. Tamassia, Checking the convexity of polytopes and the planarity of subdivisions, Comput. Geom. 11 (3–4) (1998) 187–208, http://dx.doi.org/10.1016/S0925-7721(98)00039-X.
[27] R. Diestel, Graph theory, in: Graduate Texts in Mathematics, vol. 173, Springer, 2012.
[28] P. Eades, N.C. Wormald, Fixed edge-length graph drawing is NP-hard, Discrete Appl. Math. 28 (2) (1990) 111–134, http://dx.doi.org/10.1016/0166-218X(90)90110-X.
[29] J. Geelen, A. Guo, D. McKinnon, Straight line embeddings of cubic planar graphs with integer edge lengths, J. Graph Theory 58 (3) (2008) 270–274, http://dx.doi.org/10.1002/jgt.20304.
[30] E.D. Giacomo, W. Didimo, G. Liotta, Radial drawings of graphs: Geometric constraints and trade-offs, J. Discrete Algorithms 6 (1) (2008) 109–124, http://dx.doi.org/10.1016/j.jda.2006.12.007.
[31] E.D. Giacomo, J. Hancl, G. Liotta, 2-colored point-set embeddings of partial 2-trees, in: R. Uehara, S. Hong, S.C. Nandy (Eds.), 15th International Conference and Workshops on Algorithms and Computation, WALCOM '21, in: Lecture Notes in Computer Science, vol. 12635, Springer, 2021, pp. 247–259, http://dx.doi.org/10.1007/978-3-030-68211-8_20.
[32] M.T. Goodrich, T. Johnson, Low ply drawings of trees and 2-trees, in: S. Durocher, S. Kamali (Eds.), Proceedings of the 30th Canadian Conference on Computational Geometry, CCCG '18, 2018, pp. 2–10, http://www.cs.umanitoba.ca/%7Ecccg2018/papers/session1A-p1.pdf.
[33] C. Gutwenger, P. Mutzel, A linear time implementation of SPQR-trees, in: J. Marks (Ed.), 8th International Symposium on Graph Drawing, GD '00, in: LNCS, vol. 1984, 2000, pp. 77–90, http://dx.doi.org/10.1007/3-540-44541-2_8.
[34] B. Hendrickson, Conditions for unique graph realizations, SIAM J. Comput. 21 (1) (1992) 65–84, http://dx.doi.org/10.1137/0221008.
[35] B. Hendrickson, The molecule problem: Exploiting structure in global optimization, SIAM J. Optim. 5 (4) (1995) 835–857, http://dx.doi.org/10.1137/0805040.
[36] B. Jackson, T. Jordán, Connected rigidity matroids and unique realizations of graphs, J. Comb. Theory B 94 (1) (2005) 1–29, http://dx.doi.org/10.1016/j.jctb.2004.11.002.
[37] C. Kim, K. Sol, Degenerate drawing of outerplanar graphs with two edge lengths, Discrete Appl. Math. 302 (2021) 178–188, http://dx.doi.org/10.1016/j.dam.2021.07.014.
[38] W. Lenhart, G. Liotta, D. Mondal, R.I. Nishat, Planar and plane slope number of partial 2-trees, in: S.K. Wismath, A. Wolff (Eds.), 21st International Symposium on Graph Drawing, GD '13, in: Lecture Notes in Computer Science, vol. 8242, Springer, 2013, pp. 412–423, http://dx.doi.org/10.1007/978-3-319-03841-4_36.

[39] G.D. Lozzo, W.E. Devanny, D. Eppstein, T. Johnson, Square-contact representations of partial 2-trees and triconnected simply-nested graphs, in: Y. Okamoto, T. Tokuyama (Eds.), 28th International Symposium on Algorithms and Computation, ISAAC '17, in: LIPIcs, vol. 92, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, pp. 24:1–24:14, http://dx.doi.org/10.4230/LIPIcs.ISAAC.2017.24.
[40] A. Lubiw, T. Miltzow, D. Mondal, The complexity of drawing a graph in a polygonal region, in: T.C. Biedl, A. Kerren (Eds.), 26th International Symposium on Graph Drawing and Network Visualization, GD '18, in: Lecture Notes in Computer Science, vol. 11282, Springer, 2018, pp. 387–401, http://dx.doi.org/10.1007/978-3-030-04414-5_28.
[41] A.A. Melkman, On-line construction of the convex hull of a simple polyline, Inform. Process. Lett. 25 (1) (1987) 11–12, http://dx.doi.org/10.1016/0020-0190(87)90086-X.
[42] S.L. Mitchell, Linear algorithms to recognize outerplanar and maximal outerplanar graphs, Inform. Process. Lett. 9 (5) (1979) 229–232, http://dx.doi.org/10.1016/0020-0190(79)90075-9.
[43] S. Moran, Y. Wolfstahl, One-page book embedding under vertex-neighborhood constraints, SIAM J. Discrete Math. 3 (3) (1990) 376–390, http://dx.doi.org/10.1137/0403034.
[44] N.B. Priyantha, A. Chakraborty, H. Balakrishnan, The cricket location-support system, in: R.L. Pickholtz, S.K. Das, R. Cáceres, J.J. Garcia-Luna-Aceves (Eds.), 6th Annual International Conference on Mobile Computing and Networking, MOBICOM '00, ACM, 2000, pp. 32–43, http://dx.doi.org/10.1145/345910.345917.
[45] S. Rengarajan, C.E.V. Madhavan, Stack and queue number of 2-trees, in: D. Du, M. Li (Eds.), First Annual International Conference on Computing and Combinatorics, COCOON '95, in: Lecture Notes in Computer Science, vol. 959, Springer, 1995, pp. 203–212, http://dx.doi.org/10.1007/BFb0030834.
[46] P. Rosenstiehl, R.E. Tarjan, Rectilinear planar layouts and bipolar orientations of planar graphs, Discrete Comput. Geom. 1 (1986) 343–353, http://dx.doi.org/10.1007/BF02187706.
[47] C. Savarese, J.M. Rabaey, J. Beutel, Location in distributed ad-hoc wireless sensor networks, in: 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings, vol. 4, 2001, pp. 2037–2040, http://dx.doi.org/10.1109/ICASSP.2001.940391.
[48] J.B. Saxe, Embeddability of weighted graphs in K-space is strongly NP-hard, in: CMU-CS-80-102, Carnegie-Mellon University, Department of Computer Science, 1980, https://books.google.it/books?id=vClAGwAACAAJ.
[49] M. Schaefer, Realizability of graphs and linkages, in: J. Pach (Ed.), Thirty Essays on Geometric Graph Theory, Springer New York, New York, NY, 2013, pp. 461–482, http://dx.doi.org/10.1007/978-1-4614-0110-0_24.
[50] R.I. Silveira, B. Speckmann, K. Verbeek, Non-crossing paths with geographic constraints, Discrete Math. Theoret. Comput. Sci. 21 (3) (2019) http://dx.doi.org/10.23638/DMTCS-21-3-15.
[51] M.M. Syslo, Characterizations of outerplanar graphs, Discrete Math. 26 (1) (1979) 47–53, http://dx.doi.org/10.1016/0012-365X(79)90060-8.
[52] R. Tamassia, Constraints in graph drawing algorithms, Constraints 3 (1) (1998) 87–120, http://dx.doi.org/10.1023/A:1009760732249.
[53] J. Valdes, R.E. Tarjan, E.L. Lawler, The recognition of series parallel digraphs, SIAM J. Comput. 11 (2) (1982) 298–313.
[54] M. Wiegers, Recognizing outerplanar graphs in linear time, in: G. Tinhofer, G. Schmidt (Eds.), Graph-Theoretic Concepts in Computer Science, WG '86, in: LNCS, vol. 246, Springer Berlin Heidelberg, 1987, pp. 165–176, http://dx.doi.org/10.1007/3-540-17218-1_57.
[55] Y. Yemini, Some theoretical aspects of position-location problems, in: 20th Annual Symposium on Foundations of Computer Science, FOCS '79, IEEE Computer Society, 1979, pp. 1–8, http://dx.doi.org/10.1109/SFCS.1979.39.