Planar Straight-line Realizations of 2-Trees with Prescribed Edge Lengths[*]

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

Roma Tre University, Rome, Italy
name.lastname@uniroma3.it

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 its NP-hardness, 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 longest path.

1 Introduction and Preliminary Results

The problem of producing drawings of graphs with geometric constraints is a core topic for Graph Drawing [3,4,5,11,14,23,24,27,34,43,45]. In this context, a classic question is the one of testing if a planar graph can be drawn planarly and straight-line with prescribed edge lengths. The study of such a question is related to several topics in computational geometry [17,41,47], rigidity theory [16,30,32], structural analysis of molecules [8,31], and sensor networks [13,38,40]. Formally, given a weighted planar graph $G = (V, E, \lambda)$, i.e., a planar graph 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*
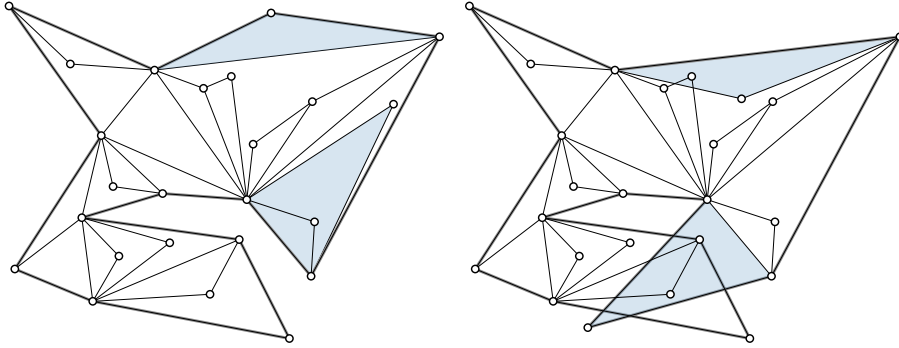

Fig. 1: A planar and a non-planar straight-line realization of the same 2-tree.

realization of G (PR for short), i.e., a planar straight-line drawing of G where the Euclidean length of each edge $e \in E$ is $\lambda(e)$. The FEPR problem was first studied by Eades and Wormald [26], who showed its NP-hardness for triconnected planar graphs and for biconnected planar graphs with unit lengths. Cabello, Demaine, and Rote strengthened this result by proving NP-hardness for triconnected planar graphs with unit lengths [12]. Abel et al. [1] proved the $\exists\mathbb{R}$-completeness of the FEPR problem with unit lengths, solving a problem posed by Schaefer [42].

Since large triconnected minors are essential in the known NP-hardness proofs of the FEPR problem, we study its complexity for 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; see Fig. 1, where we show a planar and a non-planar realization of a weighted 2-tree. Every 2-tree is planar and biconnected, and the class of 2-trees is the class of maximal series-parallel graphs. There is a vast amount of research on 2-trees in Graph Drawing (e.g., in [18,25,28,33,39]). The edge lengths of 2-trees have been studied in [9,10].

In this paper, we first show that the FEPR problem can be solved in linear time for 2-trees with prescribed embedding[1]. We note the FEPR problem is NP-hard for general planar graphs with a prescribed embedding [12]. Second, we show that, in the variable embedding setting, the FEPR problem is NP-hard when the number of distinct lengths is at least four, whereas it is linear-time solvable when the number of distinct lengths is one or two. Note that, for general planar graphs, the problem is NP-hard even when all the edges are required to have the same length [26]. Third, we deal with maximal outerplanar graphs. 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 2-trees, parameterized by the length of the longest path.

---

[1] As in [12], our algorithms adopt the real RAM model, which is customary in computational geometry and supports standard arithmetic operations in constant time.
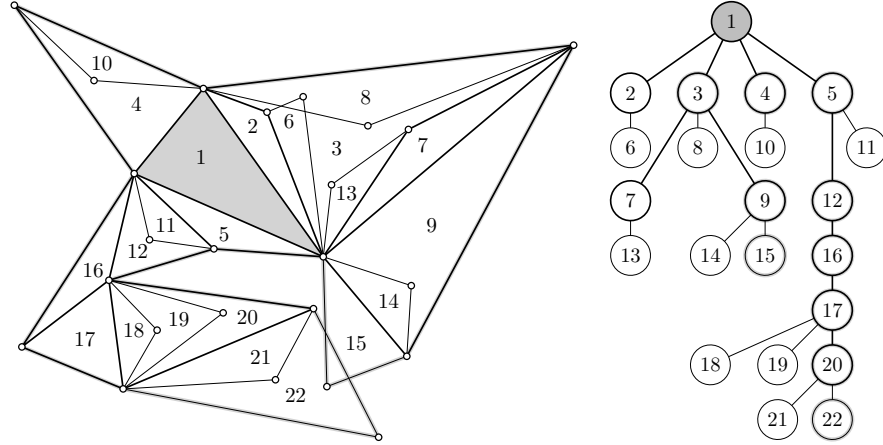
Fig. 2: A 2-tree and its decomposition tree rooted at the 3-cycle with label 1.

Because of space limitations, several proofs are omitted. They can be found in the full version of the paper [2].

Preliminaries. We assume familiarity with Graph Drawing (see, e.g., [21]). A planar drawing of a graph 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 respects $\mathcal{E}$; sometimes, we explicitly stress this.

An *outerplanar drawing* is a planar drawing in which all the vertices are incident to the outer face. An *outerplane embedding* is an equivalence class of outerplanar drawings. An *outerplanar graph* is a graph that admits an outerplanar drawing. 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. An *outerpillar* is a biconnected outerplanar graph whose dual tree is a caterpillar.

A *2-tree* is recursively defined as follows. A 3-cycle is a 2-tree. Given a 2-tree G containing an 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. We observe that the neighbors of any degree-2 vertex are adjacent. The tree-like structure of a 2-tree G is encoded by means of the *decomposition tree* T *rooted at* a 3-cycle of G. Each node in T represents a 3-cycle of G, and two nodes are adjacent if their corresponding 3-cycles share an edge; see Fig. 2. The decomposition tree of a 2-tree is easily

computed in linear time. We adopt the Euclidean metric and assume that the length function of G is such that every 3-cycle satisfies the triangle inequality. This is a necessary condition for the existence of a *straight-line realization* of G, i.e., a (not necessarily planar) drawing of G in which each edge is represented by a line segment with the prescribed length. We often refer to 3-cycles of G, nodes of T, and triangles in a straight-line realization of G interchangeably.

Prescribed Embedding. First, we deal with 2-trees with a prescribed rotation system or embedding. We start by presenting a geometric tool.

Theorem 1. *Let G be an n-vertex weighted 2-tree, $\mathcal{E}$ be a plane embedding (resp. $\mathcal{R}$ be a rotation system) for G, and Γ be a straight-line realization of G. There is an $O(n)$-time algorithm to test whether Γ is a PR respecting $\mathcal{E}$ (resp. $\mathcal{R}$).*

The proof of Theorem 1 is based on an algorithm that: **i.** tests if Γ respects $\mathcal{E}$ (resp. $\mathcal{R}$), **ii.** triangulates the faces of Γ and checks if they are simple polygons [15], and **iii.** tests if the obtained drawing is a convex subdivision [20].
We now present our prescribed embedding result.

Theorem 2. *Let G be an n-vertex weighted 2-tree and $\mathcal{E}$ be a plane embedding (resp. $\mathcal{R}$ be a rotation system) for G. There is an $O(n)$-time algorithm to test whether G admits a PR that respects $\mathcal{E}$ (resp. $\mathcal{R}$) and to construct one, if any.*

The proof of Theorem 2 is based on: **i.** computing a decomposition tree T rooted at the 3-cycle c of G with the largest sum of the edge lengths; **ii.** computing a candidate PR Γ by visiting T in pre-order while greedily adding to Γ the drawing of each 3-cycle t, by exploiting $\mathcal{E}$ (resp. $\mathcal{R}$) and the containment relationship between c and t; and **iii.** testing whether Γ is a PR of G whose plane embedding (resp. rotation system) is $\mathcal{E}$ (resp. $\mathcal{R}$) by means of Theorem 1.

2 NP-hardness for 2-trees with 4 Edge Lengths

We sketch a reduction from the NP-complete PLANAR MONOTONE 3-SAT problem [7] (PMS for short) to the FEPR problem with *four* edge lengths.

Theorem 3. *The FEPR problem is NP-hard for weighted 2-trees, even for instances whose number of distinct edge lengths is 4.*

A Boolean CNF formula ϕ is an instance of PMS if the variable-clause incidence graph G_ϕ of ϕ is planar, and each clause of ϕ is either *positive* (it consists of positive literals) or *negative* (it consists of negated literals). The PMS problem is NP-complete even when G_ϕ comes with a *monotone rectilinear representation* [7], i.e., a crossing-free drawing Γ_ϕ of G_ϕ in which **i.** variables and clauses are boxes, **ii.** edges are vertical segments, and **iii.** positive (resp. negative) clauses lie above (resp. below) the horizontal strip containing the variable boxes; see Fig. 3(a).
First, we transform Γ_ϕ into a representation Γ_ϕ^* of G_ϕ that uses segments with slope $0°$, $60°$, or $90°$; see Fig. 3(b). Then we obtain from Γ_ϕ^* a weighted
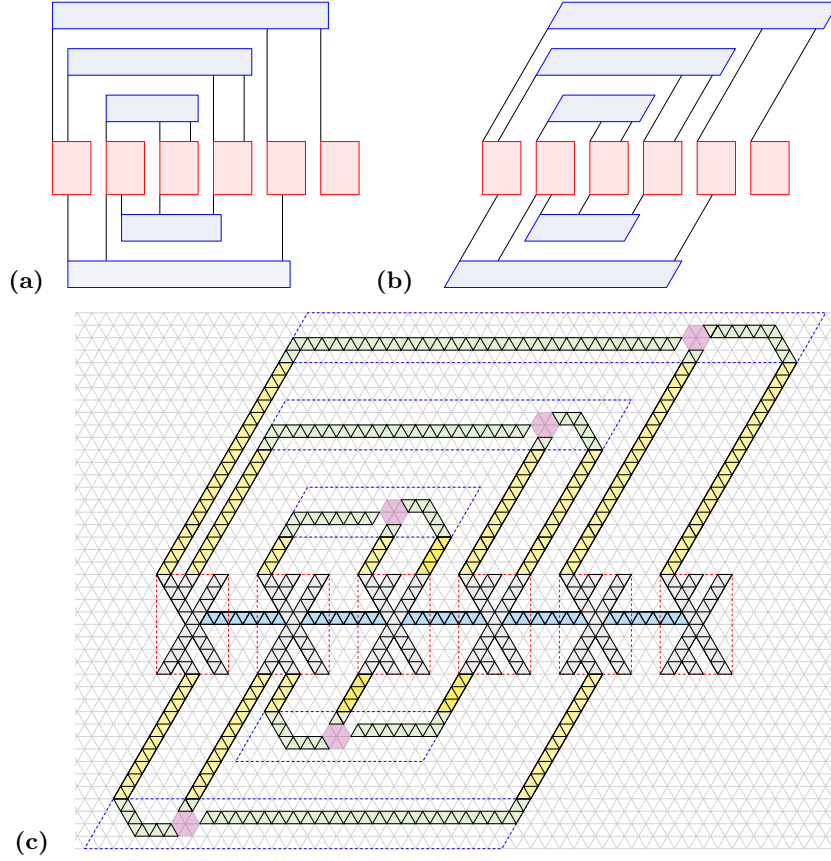
Fig. 3: (a) The monotone rectilinear representation Γ_ϕ of G_ϕ, and (b) its modified version Γ_ϕ^*. (c) Overview of the reduction showing only the frame triangles.

2-tree H_ϕ that admits a PR if and only if ϕ is satisfiable; see Fig. 3(c). The edges of H_ϕ are assigned the lengths $w_1 = 1$, $w_2 = 0.9$, $w_3 = 0.2$, and $w_4 = 1.61$. To obtain H_ϕ we construct gadgets for the variables, the clauses, and the edges of G_ϕ. Our gadgets exploit two main types of triangles: equilateral triangles with sides of length w_1 (*frame triangles*), and isosceles triangles with base of length w_1 and two sides of length w_2 (*transmission triangles*). The union of the frame triangles of the gadgets representing variables (gray), edges (yellow), and clauses (green), together with a set of frame triangles connecting the variable gadgets (blue), forms a maximal outerplanar graph. Since this graph is formed by frame triangles, it has a unique PR up to rigid transformations.

Our strategy is to construct H_ϕ from a "rigid" part (mainly formed by the union of the frame triangles of the gadgets), and a part that instead allows for different embedding choices (mainly encoded by the flips of transmission triangles). Consider for example Fig. 4, where we illustrate the PR of a clause
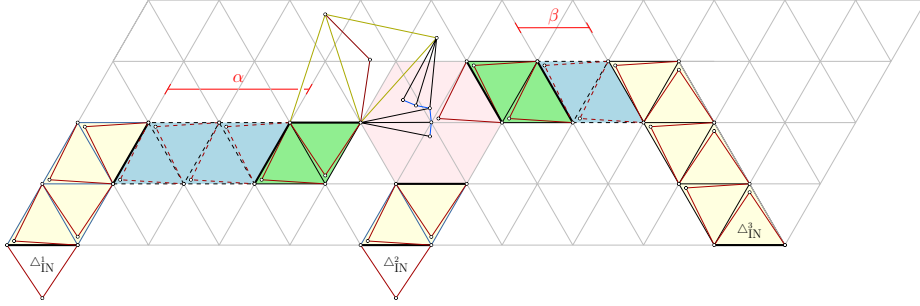
Fig. 4: The (α, β)-clause gadget with $\alpha = 4$ and $\beta = 2$. The values of α and β depend on the relative positions in Γ_ϕ^* of the involved variable gadgets.

gadget, which is the most critical gadget in the construction. Each transmission triangle $\triangle$ (in red) has two possible different embeddings. The choice of this embedding influences the choice of the embeddings of the transmission triangles that "conflict" with $\triangle$, that is, that overlaps with $\triangle$ in one of their embeddings. These chains of conflict relationships allow for "truth values" that come from the variable gadgets to "move" along the gadgets representing edges. The relevant triangles $\triangle_{\mathrm{IN}}^1$, $\triangle_{\mathrm{IN}}^2$, and $\triangle_{\mathrm{IN}}^3$ encode such values: In Fig. 4, $\triangle_{\mathrm{IN}}^1$ and $\triangle_{\mathrm{IN}}^2$ point downward since the corresponding variable is True, and $\triangle_{\mathrm{IN}}^3$ points upward since the corresponding variable is False. A special set of transmission triangles, whose flip depends on the orientation of the relevant triangles, overlap in the pink hexagonal region if all the relevant triangles point upward. Conversely, if at least one relevant triangle points downward, the clause gadget admits a PR.

3   Linear-time Algorithm for 2-trees with 2 Edge Lengths

This section is devoted to sketch the proof of the following theorem.

Theorem 4. *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 is an* $O(n)$-*time algorithm to test whether* G *admits a PR and to construct one, if any.*

Let Γ be a PR of G. If $w_1 = w_2$, then the existence of Γ implies that G is an outerplanar graph, which is a linear-time testable property [19,36,46], and that Γ is outerplanar. Since G is 2-connected, it has a unique outerplane embedding $\mathcal{E}$ which can be constructed in linear time [19,36,37,44,46]. Hence, the problem reduces to the problem of testing whether G has a PR that respects $\mathcal{E}$, which can be solved in linear time by Theorem 2.

Consider now the case in which $w_1 \neq w_2$. W.l.o.g. assume $w_1 < w_2$. Also, let $r = \frac{w_2}{w_1} > 1$. The realization of any 3-cycle of G is one of the following types of triangles: **i.** an equilateral (*small equilateral*) triangle of side w_1, **ii.** an equilateral (*big equilateral*) triangle of side w_2, **iii.** an isosceles (*tall isosceles*) triangle with
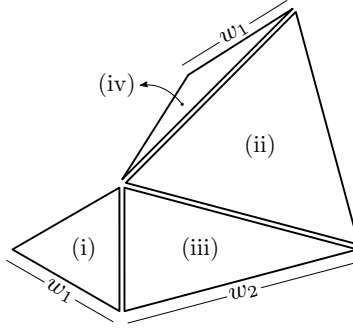
Fig. 5: The four possible types of triangles that represent the 3-cycles of G.

base w_1 and two sides of length w_2, and **iv.** an isosceles (*flat isosceles*) triangle with base w_2 and two sides of length w_1; refer to Fig. 5.

Let $\triangle_1$ and $\triangle_2$ be two triangles realizing two different 3-cycles in a PR of G. We say that $\triangle_1$ is *drawn inside* $\triangle_2$ if all the points of $\triangle_1$ are points of $\triangle_2$ and at least one vertex of $\triangle_1$ is an interior point of $\triangle_2$. Let T be the decomposition tree of G. We have the following.

Lemma 1. *If T is rooted at the 3-cycle with the largest sum of edge lengths and $\triangle_1$ is drawn inside $\triangle_2$, then $\triangle_1$ is a leaf triangle that shares a side with $\triangle_2$.*

By Lemma 1, we assume that T is rooted at a 3-cycle with the largest sum of edge lengths. The *framework* of G is the subgraph $G_F \subseteq G$ obtained, in linear time, as follows: For each leaf triangle $\triangle_i$ that can be drawn inside its parent or a sibling triangle $\triangle_j$, we remove from G the vertex v that $\triangle_i$ does not share with $\triangle_j$, along with the two edges incident to v. Note that **i.** G_F is a 2-tree which may contain any type of triangle, **ii.** by Lemma 1, no triangle of G_F is drawn inside any other triangle in any PR of G, and **iii.** T is rooted at a triangle of the framework. We test in linear time if G_F is outerplanar and, in such case, we compute in linear time its unique outerplanar embedding $\mathcal{E}$. Exploiting Theorem 2, we test if G_F admits a PR respecting $\mathcal{E}$. In the negative case G admits no PR, otherwise we denote by Γ_F the obtained PR of G_F. Hereafter, we assume that Γ_F exists.

Refer to Fig. 6, where we show an example of a PR of a weighted 2-tree. Let $L_\triangle$ be the set of leaf triangles that were removed from G to obtain G_F. Observe that $L_\triangle$ is formed by small equilateral and flat isosceles triangles. We show how to extend Γ_F to a PR of G by embedding the triangles of $L_\triangle$, if possible. Let $\triangle$ denote a triangle in $L_\triangle$. The triangle $\triangle$ has a side in G_F that is incident to either two internal faces, or to an internal face and the outer face of Γ_F; hence $\triangle$ has exactly two embedding choices. We say that $\triangle$ has an *internal embedding* if it is embedded inside an internal face of Γ_F, and an *outer embedding* otherwise. We say that $\triangle$ induces a *framework conflict* if it has an outer embedding and $\Gamma_F \cup \triangle$ is not planar; e.g., see the (dashed) outer embedding of $\triangle_{16}$ in Fig. 6. Let $\triangle_i$ and $\triangle_j$ be two triangles of $L_\triangle$. We say that $\triangle_i$ and $\triangle_j$ induce an *internal*

conflict if both have an internal embedding and $\Gamma_F \cup \triangle_i \cup \triangle_j$ is not planar; e.g., see the (dashed) internal embedding of $\triangle_9$ and the (solid) internal embedding of $\triangle_{18}$. On the other hand, we say that $\triangle_i$ and $\triangle_j$ induce an *external conflict* if both have an outer embedding and $\Gamma_F \cup \triangle_i \cup \triangle_j$ is not planar; e.g., see the (dashed) outer embeddings of $\triangle_6$ and $\triangle_{13}$.
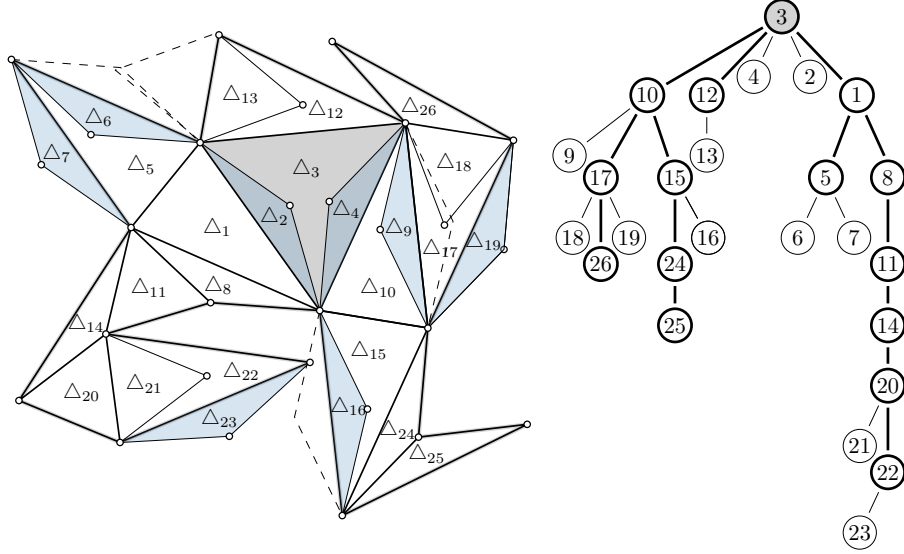

Fig. 6: (Left) A PR of a weighted 2-tree G. Dashed triangles represent alternative embeddings for some 3-cycles, which would generate conflicts. (Right) The decomposition tree of G rooted at the framework triangle $\triangle_3$. Triangles of Γ_F are thick, and edges of the leaf triangles of $L_\triangle$ that are not in Γ_F are thin.

Lemma 2. *Let $\triangle_i$ and $\triangle_j$ be two leaf triangles of $L_\triangle$ that can both be drawn inside some triangle $\triangle \in \Gamma_F$. The triangles $\triangle_i$ and $\triangle_j$ induce an internal conflict if at least one of the following properties holds true:*

(a) $\triangle_i$ and $\triangle_j$ share an edge;
(b) $\sqrt{3} < r \le 2\cos(\pi/12)$ and $\triangle$ is a tall isosceles;
(c) $1 < r \le \sqrt{3}$.

The weighted 2-tree shown in Fig. 6 is such that $\sqrt{3} < r \le 2\cos(\pi/12)$, that is $r < 2$. By Property b of Lemma 2, two flat isosceles triangles can be drawn inside a big equilateral triangle without inducing conflicts, and any pair of leaf triangles (i.e., either two flat isosceles triangles or a flat isosceles triangle and a small equilateral triangle) induce an internal conflict inside a tall isosceles triangle. On the other hand, by Property a of Lemma 2 and the fact every triangle in $L_\triangle$ has two embedding choices, for a PR of G to exist, it must hold

that no three triangles in $L_\triangle$ share the same edge with a triangle of G_F. If $L_\triangle$ satisfies this requirement, then we say that $L_\triangle$ is *consistent*.

Lemma 3. *If $L_\triangle$ is consistent, then there are $O(1)$ pairs of triangles sharing an edge with the same triangle of G_F that induce an internal conflict.*

Proof. By Lemma 1, two triangles of $L_\triangle$ that induce an internal conflict share an edge with a common triangle $\triangle$ of G_F. Since $L_\triangle$ is consistent, there exist at most 6 triangles in $L_\triangle$ incident to $\triangle$. Hence, at most $\binom{6}{2} = 15$ pairs of triangles sharing an edge with $\triangle$ can induce an internal conflict. □

Extending Γ_F to a PR of G. Next, we show how to test whether there is a choice of embeddings for the triangles in $L_\triangle$ that yields a PR of G. We distinguish two cases, based on whether $r \geq 2$ or $r < 2$.

$\boxed{\textbf{Case } \mathbf{r \geq 2.}}$ In this case there are no flat isosceles triangles, and hence the setting is much simpler than the one depicted in Fig. 6. Every leaf triangle in $L_\triangle$ is a small equilateral triangle whose parent is a tall isosceles triangle. We act as follows. For any two triangles $\triangle_1, \triangle_2 \in L_\triangle$ that share an edge e, we embed $\triangle_1$ and $\triangle_2$ in Γ_F on opposite sides of e. We embed every other leaf triangle in $L_\triangle$ inside its parent triangle. At the end of this process we obtain, in linear time, a straight-line realization Γ of G, and a plane embedding $\mathcal{E}$ of G.

Lemma 4. *G has a PR if and only if Γ is planar.*

Proof sketch. First, if two leaf triangles share an edge e, then they lie on opposite sides of e in any PR, since they would overlap otherwise. Second, each isosceles triangle may contain only one leaf triangle, since it has only one side of length w_1. Hence, embedding a leaf triangle $\triangle$ inside an isosceles triangle that shares an edge with $\triangle$ cannot cause crossings, since $\triangle$ does not induce internal conflicts. Therefore, a crossing in Γ can only be caused by framework and external conflicts, which are, however, unavoidable. □

By Lemma 4, in order to test whether G admits a PR, we can apply Theorem 1 to test in $O(n)$ time whether Γ is a PR of G with embedding $\mathcal{E}$.

$\boxed{\textbf{Case } \mathbf{r < 2.}}$ In this case there might be flat isosceles triangles in G which might or might not need to be embedded inside a framework triangle; in Fig. 6 such triangles are shaded blue. Also, more than one leaf triangle can be drawn inside the same framework triangle and this might or might not induce internal conflicts. Recall that the triangles in $L_\triangle$ are small equilateral triangles and/or flat isosceles triangles.

We construct a 2SAT formula ϕ with a Boolean variable for each triangle $\triangle \in L_\triangle$. The values of ϕ are associated with the two possible embeddings of $\triangle$. If two triangles in $L_\triangle$ induce a conflict for certain embeddings, then ϕ contains a clause that is `True` if and only if, at least one of the variables representing the triangles does not have the value corresponding to the embedding generating the conflict. Further, for each triangle that induces a framework conflict, ϕ contains

a clause that is `True` if and only if, the variable representing the triangle does not have the value corresponding to an outer embedding. We test in $O(|\phi|)$ time if ϕ is satisfiable [6]. In the positive case, we obtain a PR of G from Γ_F by embedding each triangle in $L_\triangle$ according to the value of the corresponding variable. We reject the instance in the negative case.

It only remains to prove that the number of conflicts (and hence the size of ϕ) is in $O(n)$ and that such conflicts can be found in $O(n)$ time. Detecting internal conflicts is fairly easy: **i.** by Lemma 1, triangles inducing an internal conflict are "close" in T (they share an edge with a common framework triangle); **ii.** by Lemma 3, there exist $O(1)$ leaf triangles sharing an edge with the same framework triangle; and **iii.** since $L_\triangle$ is consistent, the maximum degree of T is bounded by a constant. Hence, by traversing T we compute in $O(n)$ time the set of $O(n)$ pairs of leaf triangles that induce an internal conflict.

Efficiently detecting external and framework conflicts is more challenging. Let $L'_\triangle$ be the subset of $L_\triangle$ composed of those triangles that are incident to external edges of Γ_F. We 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 $G'_F := G_F \cup L'_\triangle$. We now construct a bounded-degree graph H whose nodes are associated with sets of vertices of G'_F so that the following properties hold: **(a)** Each node is associated with $O(1)$ degree-2 vertices of G that belong to triangles in $L'_\triangle$, **(b)** if two triangles in $L'_\triangle$ induce an external conflict, then their degree-2 vertices are associated either with the same node or with adjacent nodes, and **(c)** if a triangle $\triangle \in L'_\triangle$ intersects an edge e of G_F (inducing a framework conflict), then the degree-2 vertex of $\triangle$ and the end-vertices of e are associated either with the same node or with adjacent nodes. After constructing H, the external and framework conflicts can be detected with a linear-time traversal of H.

The graph H is defined as follows. Assume that the bottom-left corner of the bounding box of Γ'_F lies on the origin of the Cartesian axes. Consider a square grid covering the plane whose grid cells have side length $3w_2$; see Fig. 7. Assign a label $l(v) = (\lfloor \frac{x(v)}{3w_2} \rfloor, \lfloor \frac{y(v)}{3w_2} \rfloor)$ to each vertex v of G'_F. Then H has a node for each label assigned to at least one vertex of G'_F, and two distinct 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 $O(n)$ edges since it has at most n nodes and maximum degree 8.

We now prove that H satisfies Property **(a)**. The number of degree-2 vertices of G that belong to triangles in $L'_\triangle$ and are associated with a node (i, j) of H, is upper bounded by the number k of framework triangles that **i.** are contained in the union of the grid cell (i, j) and its 8 surrounding grid cells, and **ii.** share an edge with a triangle in $L'_\triangle$. Note that k is actually the number of big equilateral and tall isosceles triangles in such nine cells. Since the area of a big equilateral or tall isosceles triangle is at least the area of a small equilateral triangle, then k is upper bounded by the ratio between the area of 9 cells, which is $81w_2^2$, and the area of a small equilateral triangle, which is $w_1^2\sqrt{3}/4$. Therefore, since $r < 2$, we have that $k \in O(r^2) \subseteq O(1)$.

We next sketch an algorithm to construct H in $O(n)$ time. The vertex set of H is constructed by removing repetitions from the set of labels $l(v)$ computed for
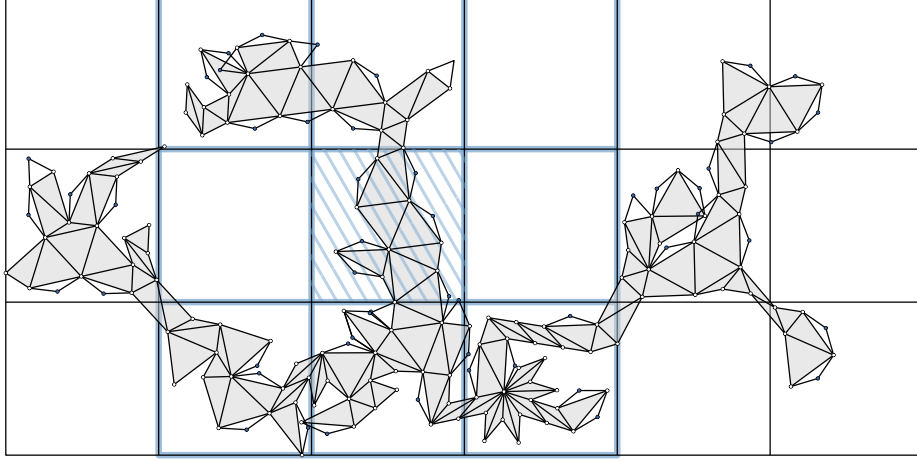
Fig. 7: The straight-line realization Γ'_F of the graph $G'_F := G_F \cup L'_\triangle$, where each triangle in $L'_\triangle$ has an outer embedding. The triangles of G_F are gray and those in $L'_\triangle$ are white. The leaf triangles in the dashed cell can induce conflicts only with the triangles in the eight highlighted cells surrounding such a cell.

the vertices v in G'_F. To this aim, we compute 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 $l(v) = (i_v, j_v)$, we have $u \prec_\pi v$ if and only if **i.** $i_u < i_v$ or **ii.** $i_u = i_v$ and $j_u < j_v$. Since G'_F is connected and any edge of G'_F has length at most w_2, then $0 \leq i, j \leq \frac{w_2 n}{3 w_2} = \frac{1}{3} n$ for any label (i, j). Hence, we compute π in $O(n)$ time with counting sort. Since vertices with the same label are consecutive in π, repetitions can be removed with a linear scan of π. The edge set of H consists of four disjoint subsets E_-, $E_|$, $E_/$, and $E_\backslash$. These sets contain the edges that connect nodes of H corresponding to grid cells that are adjacent horizontally, vertically, along the main, and the minor diagonal, respectively; see Fig. 8. We appropriately define four orders π_-, $\pi_|$, $\pi_/$, and $\pi_\backslash$ of the nodes of H such that nodes that are connected by an edge in E_-, $E_|$, $E_/$, and $E_\backslash$ are consecutive in the corresponding order. We compute the four sets of edges with a linear scan of the orders π_-, $\pi_|$, $\pi_/$, and $\pi_\backslash$.

4   Maximal Outerplanar Graphs

In this section we study the FEPR problem for weighted outerplanar 2-trees, i.e., for weighted maximal outerplanar graphs. We prove the following theorems.

Theorem 5. *Let G be an n-vertex weighted maximal outerpath. There is an $O(n)$-time algorithm to test whether G admits a PR and to construct one, if any.*

Theorem 6. *Let G be an n-vertex weighted maximal outerpillar. There is an $O(n^3)$-time algorithm to test whether G admits a PR and to construct one, if any.*
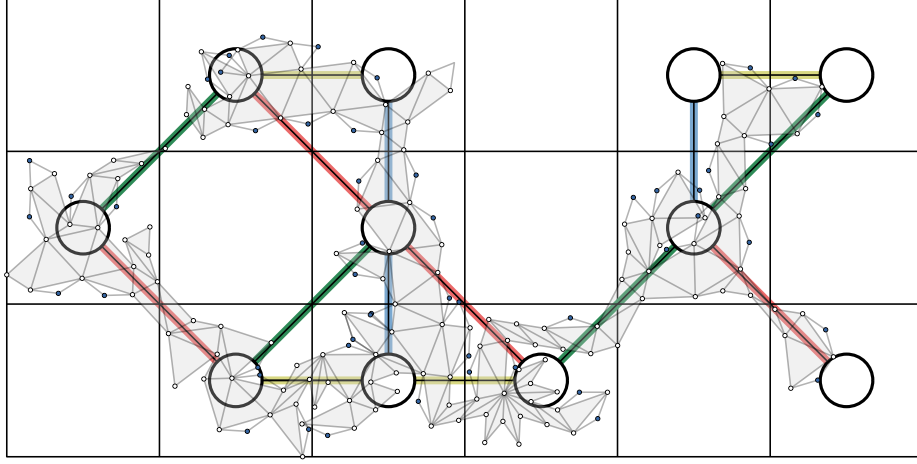
Fig. 8: The edge set of the graph H computed from the drawing Γ'_F in Fig. 7. The sets $E_-, E_|, E_/$, and $E_\backslash$ are shown in yellow, blue, green, and red, respectively.

Let G be a weighted 2-tree and e be an edge of G. An *e-outer realization* of G is a PR of G such that e is incident to the outer face. An e-outer realization Γ of G is *e-optimal* if, for every e-outer realization Γ' of G, there is a rigid transformation of Γ such that the segment representing e coincides with the one in Γ' and such that the interior of Γ is a subset of the interior of Γ'.

We sketch the proof of Theorem 5; the proof of Theorem 6 uses similar ideas. Let G be an n-vertex weighted maximal outerpath; see Fig. 9. Let T be the dual tree of the outerplane embedding $\mathcal{O}$ of G; since G is an outerpath, T is a path $(p_1, \ldots, p_k)$. For $i = 1, \ldots, k$, let c_i be the 3-cycle of G bounding the internal face of $\mathcal{O}$ dual to p_i and let $\mathcal{C}_i$ be the unique, up to rigid transformation, PR of c_i. For $i = 1, \ldots, k-1$, let e_i be the edge of G dual to (p_i, p_{i+1}). Let $x \in \{1, \ldots, k\}$
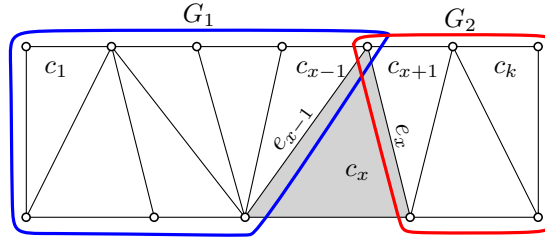

Fig. 9: A maximal outerpath G.

be such that c_x has maximum edge length sum. Let G_1 and G_2 be the subgraphs of G composed of the cycles $c_1, c_2, \ldots, c_{x-1}$ and $c_{x+1}, c_{x+2}, \ldots, c_k$, respectively.

Since the length of c_x is maximum, the restrictions of any PR of G to G_1 and G_2 are e_{x-1}-outer and e_x-outer realizations, respectively. We prove that G_1 (resp. G_2) admits an e_{x-1}-outer (resp. e_x-outer) realization if and only if it admits an e_{x-1}-optimal (resp. e_x-optimal) realization. The core of the proof of Theorem 5 is an $O(n)$-time algorithm, called OUTER-CHECKER, that constructs an e_{x-1}-optimal (resp. an e_x-optimal) realization Γ_1 of G_1 (resp. Γ_2 of G_2) and its plane embedding, if any such a realization exists. If OUTER-CHECKER concludes that both G_1 and G_2 admit a PR, then Γ_1 and Γ_2 (as well as their embeddings) can be combined in four ways with $\mathcal{C}_x$ (see Fig. 10) and each resulting straight-line realization can be tested for planarity in $O(n)$ time, by Theorem 1.
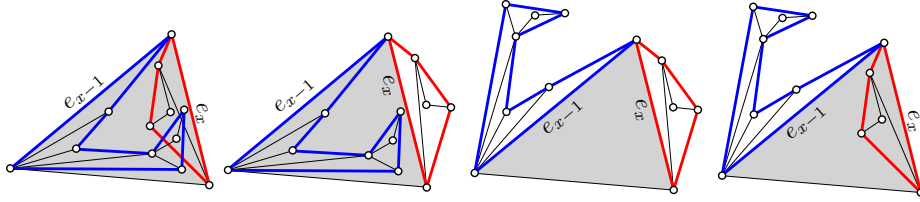

Fig. 10: The four different ways to combine Γ_1 and Γ_2 with $\mathcal{C}_x$.

We describe how OUTER-CHECKER works on G_1. A key observation is that the restriction of any e_{x-1}-optimal realization of G_1 to the graph G_1^i composed of the cycles $c_1, c_2, \ldots, c_i$ is an e_i-optimal realization of G_1^i. This allows OUTER-CHECKER to work by induction on i to decide whether G_1^i has an e_i-optimal realization Γ_1^i. If $i = 1$, the graph G_1^1 is the cycle c_1 whose unique PR $\mathcal{C}_1$ is e_1-optimal. If $i > 1$, an e_i-optimal realization Γ_1^i of G_1^i is constructed, if it exists, by combining Γ_1^{i-1} and $\mathcal{C}_i$ so that e_{i-1} coincides in the two realizations. Three things might happen. First, if Γ_1^{i-1} "fits" inside $\mathcal{C}_i$, as in Fig. 11(left), then the resulting PR Γ_1^i is e_i-optimal. Else, if Γ_1^{i-1} "fits" outside $\mathcal{C}_i$, as in Fig. 11(middle), once cycles $\mathcal{C}_i$ and $\mathcal{C}_{i-1}$ lie on different sides of e_{i-1}, then the resulting PR Γ_1^i is e_i-optimal. Otherwise, G_1^i admits no e_i-optimal realization, as in Fig. 11(right).
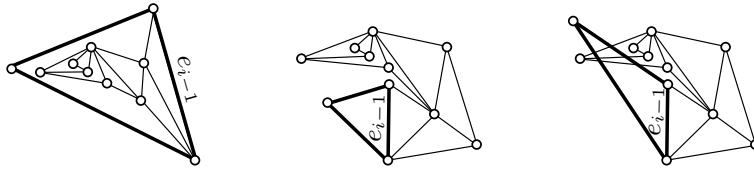

Fig. 11: The three cases in the construction of Γ_1^i. The triangle $\mathcal{C}_i$ is bold. (Left) Γ_1^{i-1} fits inside $\mathcal{C}_i$. (Middle) Γ_1^{i-1} does not fit inside $\mathcal{C}_i$, but it fits outside $\mathcal{C}_i$. (Right) Γ_1^{i-1} fits neither inside nor outside $\mathcal{C}_i$.

A naive implementation of Outer-Checker takes $O(n^2)$ time. Indeed, for each of the $O(n)$ inductive steps, one can check in $O(n)$ time whether Γ_1^{i-1} fits inside and/or outside $\mathcal{C}_i$ using Theorem 1. We achieve $O(n)$ total running time avoiding a planarity test at each step. For $i = 1, \ldots, x - 1$, we compute a "candidate" straight-line realization Γ_1^i of G_1^i, and only test for planarity the final realization Γ_1^{x-1}. By "candidate" we mean that, if G_1^i admits an e_i-optimal realization, then Γ_1^i is such a realization. In order to do that, Outer-Checker dynamically maintains the boundary $\mathcal{B}_1^i$ of the convex hull of Γ_1^i, which is guaranteed to actually be the boundary of the convex hull of Γ_1^i if Γ_1^i is planar. We compute $\mathcal{B}_1^i$ by suitably exploiting a linear-time algorithm by Melkman [35], which incrementally computes the convex hull of a point set spanned by a planar path, provided that the points are given in the order of the path.

After constructing Γ_1^{x-1} (which comes with a plane embedding), we test its planarity in $O(n)$ time using Theorem 1. If the test is successful, Γ_1^{x-1} is an e_{x-1}-optimal PR of G_1^{x-1}, otherwise no e_{x-1}-optimal PR of G_1^{x-1} exists. Each step of Outer-Checker takes $O(1)$ time, except for the computation of the boundary $\mathcal{B}_1^i$. However, the computation of the boundaries $\mathcal{B}_1^1, \mathcal{B}_1^2, \ldots, \mathcal{B}_1^{x-1}$ takes $O(n)$ time in total [35]. Hence, the overall running time of Outer-Checker is in $O(n)$.

5   2-Trees with Short Longest Path

In this section, we sketch a proof of the following theorem.

Theorem 7. *Let G be an n-vertex weighted 2-tree and let ℓ be the length of a longest path of G. There is an $n^{O(4^\ell)}$-time algorithm to test whether G admits a PR and to construct one, if any.*

Theorem 7 is actually a corollary of a stronger theorem, which relates to SPQ-trees; these are a specialization for 2-trees of the well-known *SPQR-trees* [22,29]. The *SPQ-tree* $\mathcal{T}$ of G is a tree that represents a recursive decomposition of G into subgraphs along separation pairs. Each node μ of $\mathcal{T}$ corresponds to a subgraph G_μ of G, which is joined to the rest of the graph via two vertices u_μ and v_μ. Assume that $\mathcal{T}$ is rooted at the neighbor of an edge of G with maximum length and let h be the height of $\mathcal{T}$. We design an $n^{O(2^h)}$-time algorithm that tests whether G admits a PR and, in the positive case, constructs such a realization. Then Theorem 7 follows, as we can prove that $h \leq 2\ell - 2$.

The $n^{O(2^h)}$-time algorithm performs a visit of $\mathcal{T}$. When visiting a node μ, the algorithm either concludes that G admits no PR, or constructs a set $\mathcal{R}_\mu$ of "optimal" PRs of G_μ. Here, "optimal" means that, for every PR Γ_μ of G_μ, there is a PR $\Gamma_\mu' \in \mathcal{R}_\mu$ whose interior is a subset of the interior of Γ_μ, after a suitable rigid transformation. The main ingredient needed for bounding the running time of the algorithm is the following. Suppose that G_μ consists of a "parallel" composition of graphs $G_{\nu_1}, \ldots, G_{\nu_k}$. Then "few" of the permutations of $G_{\nu_1}, \ldots, G_{\nu_k}$ need to be considered when constructing $\mathcal{R}_\mu$. Namely, we can sort $G_{\nu_1}, \ldots, G_{\nu_k}$ by increasing length of the 2-edge paths between u_μ and v_μ

they contain. Then, in any PR of G, the graph G_{ν_i} is either "to the left" or "to the right" of all the graphs $G_{\nu_1}, \ldots, G_{\nu_{i-1}}$; further, whether a PR of G is optimal only depends on the choice of the "leftmost" and "rightmost" graphs among $G_{\nu_1}, \ldots, G_{\nu_k}$ (and on their drawings, which are taken from $\mathcal{R}_{\nu_1}, \ldots, \mathcal{R}_{\nu_k}$), and not on the permutation of the remaining graphs, as long as planarity is ensured.

6 Open problems

Our results on the FEPR problem when G is a 2-tree motivate the study of several open questions:

- Determine the computational complexity of the FEPR problem for weighted 2-trees with 3 prescribed edge lengths (we proved it is linear-time solvable for 2 and NP-hard for 4).
- Determine if it is possible to improve our XP algorithm for general 2-trees to an FPT algorithm.
- Study the computational complexity of the FEPR problem for general maximal outerplanar graphs.
- Study the computational complexity of the FEPR problem for graphs with treewidth 2 and for 2-degenerate planar graphs; both these classes generalize the one of 2-trees.

References

1. Abel, Z., Demaine, E.D., Demaine, M.L., Eisenstat, S., Lynch, J., Schardl, T.B.: Who needs crossings? Hardness of plane graph rigidity. In: Fekete, S.P., Lubiw, A. (eds.) 32nd International Symposium on Computational Geometry (SoCG '16). LIPIcs, vol. 51, pp. 3:1–3:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2016). https://doi.org/10.4230/LIPIcs.SoCG.2016.3
2. Alegría, C., Borrazzo, M., Da Lozzo, G., Di Battista, G., Frati, F., Patrignani, M.: Planar straight-line realizations of 2-trees with prescribed edge lengths. CoRR **abs/2108.09483** (2021), https://arxiv.org/abs/2108.09483
3. Alon, N., Feldheim, O.N.: Drawing outerplanar graphs using three edge lengths. Comput. Geom. **48**(3), 260–267 (2015). https://doi.org/10.1016/j.comgeo.2014.10.006
4. Angelini, P., Da Lozzo, G., Di Bartolomeo, M., Di Battista, G., Hong, S., Patrignani, M., Roselli, V.: Anchored drawings of planar graphs. In: Duncan, C.A., Symvonis, A. (eds.) 22nd International Symposium on Graph Drawing (GD '14). Lecture Notes in Computer Science, vol. 8871, pp. 404–415. Springer (2014). https://doi.org/10.1007/978-3-662-45803-7_34
5. Angelini, P., Da Lozzo, G., Di Battista, G., Di Donato, V., Kindermann, P., Rote, G., Rutter, I.: Windrose planarity: Embedding graphs with direction-constrained edges. ACM Trans. Algorithms **14**(4), 54:1–54:24 (2018). https://doi.org/10.1145/3239561
6. Aspvall, B., Plass, M.F., Tarjan, R.E.: A linear-time algorithm for testing the truth of certain quantified boolean formulas. Information Processing Letters **8**(3), 121–123 (1979). https://doi.org/10.1016/0020-0190(79)90002-4

7. de Berg, M., Khosravi, A.: Optimal binary space partitions for segments in the plane. Int. J. Comput. Geom. Appl. **22**(3), 187–206 (2012), http://www.worldscinet.com/doi/abs/10.1142/S0218195912500045

8. Berger, B., Kleinberg, J.M., Leighton, F.T.: Reconstructing a three-dimensional model with arbitrary errors. J. ACM **46**(2), 212–235 (1999). https://doi.org/10.1145/301970.301972

9. Blazej, V., Fiala, J., Liotta, G.: On the edge-length ratio of 2-trees. In: Auber, D., Valtr, P. (eds.) 28th International Symposium on Graph Drawing and Network Visualization (GD '20). Lecture Notes in Computer Science, vol. 12590, pp. 85–98. Springer (2020). https://doi.org/10.1007/978-3-030-68766-3_7

10. Borrazzo, M., Frati, F.: On the planar edge-length ratio of planar graphs. J. Comput. Geom. **11**(1), 137–155 (2020), https://journals.carleton.ca/jocg/index.php/jocg/article/view/470

11. Brandes, U., Schlieper, B.: Angle and distance constraints on tree drawings. In: Kaufmann, M., Wagner, D. (eds.) 14th International Symposium on Graph Drawing (GD '06). Lecture Notes in Computer Science, vol. 4372, pp. 54–65. Springer (2006). https://doi.org/10.1007/978-3-540-70904-6_7

12. Cabello, S., Demaine, E.D., Rote, G.: Planar embeddings of graphs with specified edge lengths. Journal of Graph Algorithms and Applications **11**(1), 259–276 (2007). https://doi.org/10.7155/jgaa.00145

13. Capkun, S., Hamdi, M., Hubaux, J.: GPS-free positioning in mobile ad hoc networks. Clust. Comput. **5**(2), 157–167 (2002). https://doi.org/10.1023/A:1013933626682

14. Chaplick, S., Kindermann, P., Löffler, A., Thiele, F., Wolff, A., Zaft, A., Zink, J.: Recognizing stick graphs with and without length constraints. J. Graph Algorithms Appl. **24**(4), 657–681 (2020). https://doi.org/10.7155/jgaa.00524

15. Chazelle, B.: Triangulating a simple polygon in linear time. Discret. Comput. Geom. **6**, 485–524 (1991). https://doi.org/10.1007/BF02574703

16. Connelly, R.: On generic global rigidity. In: Gritzmann, P., Sturmfels, B. (eds.) Proceedings of a DIMACS Workshop on Applied Geometry And Discrete Mathematics. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 4, pp. 147–156. DIMACS/AMS (1990). https://doi.org/10.1090/dimacs/004/11

17. Coullard, C.R., Lubiw, A.: Distance visibility graphs. Int. J. Comput. Geom. Appl. **2**(4), 349–362 (1992). https://doi.org/10.1142/S0218195992000202

18. Da Lozzo, G., Devanny, W.E., Eppstein, D., Johnson, T.: Square-contact representations of partial 2-trees and triconnected simply-nested graphs. In: Okamoto, Y., Tokuyama, T. (eds.) 28th International Symposium on Algorithms and Computation (ISAAC '17). LIPIcs, vol. 92, pp. 24:1–24:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2017). https://doi.org/10.4230/LIPIcs.ISAAC.2017.24

19. Deng, T.: On the implementation and refinement of outerplanar graph algorithms. Master's thesis, University of Windsor, Ontario, Canada (2007)

20. Devillers, O., Liotta, G., Preparata, F.P., Tamassia, R.: Checking the convexity of polytopes and the planarity of subdivisions. Comput. Geom. **11**(3-4), 187–208 (1998). https://doi.org/10.1016/S0925-7721(98)00039-X

21. Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for the Visualization of Graphs. Prentice-Hall (1999)

22. Di Battista, G., Tamassia, R.: On-line planarity testing. SIAM J. Comput. **25**(5), 956–997 (1996). https://doi.org/10.1137/S0097539794280736

23. Di Battista, G., Tamassia, R., Tollis, I.G.: Constrained visibility representations of graphs. Inf. Process. Lett. **41**(1), 1–7 (1992). https://doi.org/10.1016/0020-0190(92)90072-4

24. Di Giacomo, E., Didimo, W., Liotta, G.: Radial drawings of graphs: Geometric constraints and trade-offs. J. Discrete Algorithms **6**(1), 109–124 (2008). https://doi.org/10.1016/j.jda.2006.12.007

25. Di Giacomo, E., Hancl, J., Liotta, G.: 2-colored point-set embeddings of partial 2-trees. In: Uehara, R., Hong, S., Nandy, S.C. (eds.) 15th International Conference and Workshops on Algorithms and Computation (WALCOM '21). Lecture Notes in Computer Science, vol. 12635, pp. 247–259. Springer (2021). https://doi.org/10.1007/978-3-030-68211-8_20

26. Eades, P., Wormald, N.C.: Fixed edge-length graph drawing is NP-hard. Disc. Appl. Math. **28**(2), 111–134 (1990). https://doi.org/10.1016/0166-218X(90)90110-X

27. Geelen, J., Guo, A., McKinnon, D.: Straight line embeddings of cubic planar graphs with integer edge lengths. J. Graph Theory **58**(3), 270–274 (2008). https://doi.org/10.1002/jgt.20304

28. Goodrich, M.T., Johnson, T.: Low ply drawings of trees and 2-trees. In: Durocher, S., Kamali, S. (eds.) Proceedings of the 30th Canadian Conference on Computational Geometry (CCCG '18). pp. 2–10 (2018), http://www.cs.umanitoba.ca/%7Ecccg2018/papers/session1A-p1.pdf

29. Gutwenger, C., Mutzel, P.: A linear time implementation of SPQR-trees. In: Marks, J. (ed.) 8th International Symposium on Graph Drawing (GD '00). LNCS, vol. 1984, pp. 77–90 (2000). https://doi.org/10.1007/3-540-44541-2_8

30. Hendrickson, B.: Conditions for unique graph realizations. SIAM J. Comput. **21**(1), 65–84 (1992). https://doi.org/10.1137/0221008

31. Hendrickson, B.: The molecule problem: Exploiting structure in global optimization. SIAM J. Optim. **5**(4), 835–857 (1995). https://doi.org/10.1137/0805040

32. Jackson, B., Jordán, T.: Connected rigidity matroids and unique realizations of graphs. J. Comb. Theory, Ser. B **94**(1), 1–29 (2005). https://doi.org/10.1016/j.jctb.2004.11.002

33. Lenhart, W., Liotta, G., Mondal, D., Nishat, R.I.: Planar and plane slope number of partial 2-trees. In: Wismath, S.K., Wolff, A. (eds.) 21st International Symposium on Graph Drawing (GD '13). Lecture Notes in Computer Science, vol. 8242, pp. 412–423. Springer (2013). https://doi.org/10.1007/978-3-319-03841-4_36

34. Lubiw, A., Miltzow, T., Mondal, D.: The complexity of drawing a graph in a polygonal region. In: Biedl, T.C., Kerren, A. (eds.) 26th International Symposium on Graph Drawing and Network Visualization (GD '18). Lecture Notes in Computer Science, vol. 11282, pp. 387–401. Springer (2018). https://doi.org/10.1007/978-3-030-04414-5_28

35. Melkman, A.A.: On-line construction of the convex hull of a simple polyline. Inf. Process. Lett. **25**(1), 11–12 (1987). https://doi.org/10.1016/0020-0190(87)90086-X

36. Mitchell, S.L.: Linear algorithms to recognize outerplanar and maximal outerplanar graphs. Inf. Process. Lett. **9**(5), 229–232 (1979). https://doi.org/10.1016/0020-0190(79)90075-9

37. Moran, S., Wolfstahl, Y.: One-page book embedding under vertex-neighborhood constraints. SIAM J. Discrete Math. **3**(3), 376–390 (1990). https://doi.org/10.1137/0403034

38. Priyantha, N.B., Chakraborty, A., Balakrishnan, H.: The cricket location-support system. In: Pickholtz, R.L., Das, S.K., Cáceres, R., Garcia-Luna-Aceves, J.J. (eds.) 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00). pp. 32–43. ACM (2000). https://doi.org/10.1145/345910.345917

39. Rengarajan, S., Madhavan, C.E.V.: Stack and queue number of 2-trees. In: Du, D., Li, M. (eds.) First Annual International Conference on Computing and Combinatorics (COCOON '95). Lecture Notes in Computer Science, vol. 959, pp. 203–212. Springer (1995). https://doi.org/10.1007/BFb0030834

40. Savarese, C., Rabaey, J., Beutel, J.: Location in distributed ad-hoc wireless sensor networks. In: 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. vol. 4, pp. 2037–2040 (2001). https://doi.org/10.1109/ICASSP.2001.940391

41. Saxe, J.: Embeddability of Weighted Graphs in K-space is Strongly NP-hard. CMU-CS-80-102, Carnegie-Mellon University, Department of Computer Science (1980), https://books.google.it/books?id=vClAGwAACAAJ

42. Schaefer, M.: Realizability of graphs and linkages. In: Pach, J. (ed.) Thirty Essays on Geometric Graph Theory, pp. 461–482. Springer New York, New York, NY (2013). https://doi.org/10.1007/978-1-4614-0110-0_24

43. Silveira, R.I., Speckmann, B., Verbeek, K.: Non-crossing paths with geographic constraints. Discret. Math. Theor. Comput. Sci. **21**(3) (2019), http://dmtcs.episciences.org/5495

44. Syslo, M.M.: Characterizations of outerplanar graphs. Discret. Math. **26**(1), 47–53 (1979). https://doi.org/10.1016/0012-365X(79)90060-8

45. Tamassia, R.: Constraints in graph drawing algorithms. Constraints An Int. J. **3**(1), 87–120 (1998). https://doi.org/10.1023/A:1009760732249

46. Wiegers, M.: Recognizing outerplanar graphs in linear time. In: Tinhofer, G., Schmidt, G. (eds.) WG '86. LNCS, vol. 246, pp. 165–176. Springer (1987)

47. Yemini, Y.: Some theoretical aspects of position-location problems. In: 20th Annual Symposium on Foundations of Computer Science (FOCS '79). pp. 1–8. IEEE Computer Society (1979). https://doi.org/10.1109/SFCS.1979.39