Level-Planar Drawings with Few Slopes

Guido Brückner$^{(\boxtimes)}$, Nadine Davina Krisam, and Tamara Mchedlidze

Karlsruhe Institute of Technology, Karlsruhe, Germany
brueckner@kit.edu, nadine.krisam@student.kit.edu, mched@iti.uka.de

Abstract. We introduce and study level-planar straight-line drawings with a fixed number λ of slopes. For proper level graphs, we give an $O(n \log^2 n / \log \log n)$-time algorithm that either finds such a drawing or determines that no such drawing exists. Moreover, we consider the partial drawing extension problem, where we seek to extend an immutable drawing of a subgraph to a drawing of the whole graph, and the simultaneous drawing problem, which asks about the existence of drawings of two graphs whose restrictions to their shared subgraph coincide. We present $O(n^{4/3} \log n)$-time and $O(\lambda n^{10/3} \log n)$-time algorithms for these respective problems on proper level-planar graphs.

We complement these positive results by showing that testing whether non-proper level graphs admit level-planar drawings with λ slopes is NP-hard even in restricted cases.

1 Introduction

Directed graphs explaining hierarchy naturally appear in multiple industrial and academic applications. Some examples include PERT diagrams, UML component diagrams, text edition networks [1], text variant graphs [19], philogenetic and neural networks. In these, and many other applications, it is essential to visualize the implied directed graph so that the viewer can perceive the hierarchical structure it contains. By far the most popular way to achieve this is to apply the *Sugyiama framework* – a generic network visualization algorithm that results in a drawing where each vertex lies on a horizontal line, called *layer*, and each edge is directed from a lower layer to a higher layer [14].

The Sugyiama framework consists of several steps: elimination of directed cycles in the initial graph, assignment of vertices to layers, vertex ordering and coordinate assignement. During each of these steps several criteria are optimized, by leading to more readable visualizations, see e.g. [14]. In this paper we concentrate on the last step of the framework, namely coordinate assignment. Thus, the subject of our study are *level graphs* defined as follows. Let $G = (V, E)$ be a directed graph. A *k-level assignment* of G is a function $\ell : V \to \{1, 2, \ldots, k\}$ that assigns each vertex of G to one of k levels. We refer to G together with ℓ as to a *(k-)level graph*. The *length* of an edge (u, v) is defined as $\ell(v) - \ell(u)$. We say that G is *proper* if all edges have length one. The level graph shown in Fig. 1(a) is proper, whereas the one shown in (b) is not. For a non-proper level graph G

there exists a *proper subdivision* obtained by subdividing all edges with length greater than one which result in a proper graph.

A *level drawing* Γ of a level graph G maps each vertex $v \in V$ to a point on the horizontal line with y-coordinate $\ell(v)$ and a real x-coordinate $\Gamma(v)$, and each edge to a y-monotone curve between its endpoints. A level drawing is called *level-planar* if no two edges intersect except in common endpoints. It is *straight-line* if the edges are straight lines. A level drawing of a proper (subdivision of a) level graph G induces a total left-to-right order on the vertices of a level. We say that two drawings are *equivalent* if they induce the same order on every level. An equivalence class of this equivalence relation is an *embedding* of G. We refer to G together with an embedding to as *embedded level graph* $\mathcal{G}$. The third step of Sugyiama framework, vertex ordering, results in an embedded level graph.

The general goal of the coordinate assignment step is to produce a final visualization while further improving its readability. The criteria of readability that have been considered in the literature for this step include straightness and verticality of the edges [14]. Here we study the problem of coordinate assignment step with bounded number of slopes. The *slope* of an edge (u, v) in Γ is defined as $(\Gamma(v) - \Gamma(u))/(\ell(v) - \ell(u))$. For proper level graphs this simplifies to $\Gamma(v) - \Gamma(u)$. We restrict our study to drawings in which all slopes are non-negative; such drawings can be transformed into drawings with negative slopes by shearing; see Fig. 1. A level drawing Γ is a λ-*slope drawing* if all slopes in Γ appear in the set $\{0, 1, \ldots, \lambda - 1\}$. We study embedding-preserving straight-line level-planar λ-slope drawings, or λ-*drawings* for short and refer to the problem of finding these drawings as λ-DRAWABILITY. Since the possible edge slopes in a λ-drawing are integers all vertices lie on the integer grid.
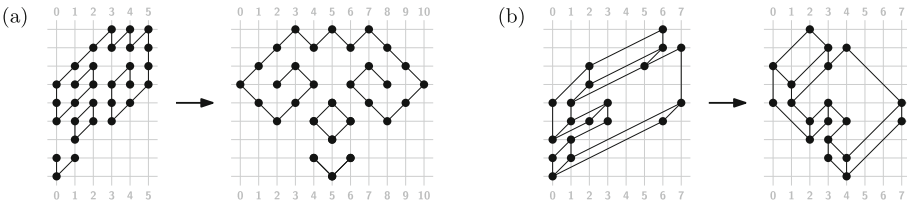

Fig. 1. Shearing drawings to change the slopes. In (a), the left drawing with slopes 0 and 1 is transformed into the right orthogonal drawing, i.e., one with slopes -1 and 1. In (b), the left drawing with slopes $0, 1$ and 2 is transformed into a drawing with slopes $-1, 0$ and 1.

Related Work. The number of slopes used for the edges in a graph drawing can be seen as an indication of the simplicity of the drawing. For instance, the measure *edge orthogonality*, which specifies how close a drawing is to an *orthogonal drawing*, where edges are polylines consisting of horizontal and vertical segments only, has been proposed as a measure of aesthetic quality of a graph

drawing [31]. In a similar spirit, Kindermann et al. studied the effect reducing the segment complexity on the aesthetics preference of graph drawings and observed that in some cases people prefer drawings using lower segment complexity [22]. More generally, the use of few slopes for a graph drawing may contribute to the formation of "Prägnanz" ("good figure" in German) of the visualization, that accordingly to the Gestalt law of Prägnanz, or law of simplicity, contributes to the perceived quality of the visualizations. This is design principle often guides the visualization of metro maps. See [28] for a survey of the existing approaches, most of which generate octilinear layouts of metro maps, and [27] for a recent model for drawing more general k-linear metro maps.

Level-planar drawing with few slopes have not been considered in the literature but drawings of undirected graphs with few slopes have been extensively studied. The (*planar*) *slope number* of a (planar) graph G is the smallest number s so that G has a (planar) straight-line drawing with edges of at most s distinct slopes. Special attention has been given to proving bounds on the (planar) slope number of undirected graph classes [4,9–12,20,21,23,24,29]. Determining the planar slope number is hard in the existential theory of reals [16].

Several graph visualization problems have been considered in the partial and simultaneous settings. In the *partial drawing extension* problem, one is presented with a graph and an immutable drawing of some subgraph thereof. The task is to determine whether the given drawing of the subgraph can be completed to a drawing of the entire graph. The problem has been studied for the planar setting [25,30] and also the level-planar setting [8]. In the *simultaneous drawing* problem, one is presented with two graphs that may share some subgraph. The task is to draw both graphs so that the restrictions of both drawings to the shared subgraph are identical. We refer the reader to [5] for an older literature overview. The problem has been considered for orthogonal drawings [2] and level-planar drawings [3]. Up to our knowledge, neither partial nor simultaneous drawings have been considered in the restricted slope setting.

Contribution. We introduce and study the λ-Drawability problem. To solve this problem for proper level graphs, we introduce two models. In Sect. 3 we describe the first model, which uses a classic integer-circulation-based approach. This model allows us to solve the λ-Drawability in $O(n \log^3 n)$ time and obtain a λ-drawing within the same running time if one exists. In Sect. 4, we describe the second distance-based model. It uses the duality between flows in the primal graph and distances in the dual graph and allows us to solve the λ-Drawability in $O(n \log^2 n / \log \log n)$ time. We also address the partial and simultaneous settings. The classic integer-circulation-based approach can be used to extend connected partial λ-drawings in $O(n \log^3 n)$ time. In Sect. 5, we build on the distance-based model to extend not-necessarily-connected partial λ-drawings in $O(n^{4/3} \log n)$ time, and to obtain simultaneous λ-drawings in $O(\lambda n^{10/3} \log n)$ time if they exist. We finish with some concluding remarks in Sect. 6 and refer to the full version [7] for a proof that 2-Drawability is NP-hard even for biconnected graphs where all edges have length one or two.

2   Preliminaries

Let Γ be a level-planar drawing of an embedded level-planar graph $\mathcal{G}$. The *width* of Γ is defined as $\max_{v \in V} \Gamma(v) - \min_{v \in V} \Gamma(v)$. An integer $\bar{x}$ is a *gap* in Γ if it is $\Gamma(v) \neq \bar{x}$ for all $v \in V$, $\Gamma(v_1) < \bar{x}$ and $\Gamma(v_2) > \bar{x}$ for some $v_1, v_2 \in V$, and $\Gamma(u) < \bar{x} < \Gamma(v)$ for no edge $(u, v) \in E$. A drawing Γ is *compact* if it has no gap. Note that a λ-drawing of a connected level-planar graph is inherently compact. While in a λ-drawing of a non-connected level-planar graph every gap can be eliminated by a shift. The fact that we only need to consider compact λ-drawings helps us to limit their width. Thus, any compact λ-drawing has a width of at most $(\lambda - 1)(n - 1)$.

Let u and v be two vertices on the same level i. With $[u, v]_\mathcal{G}$ (or $[u, v]$ when $\mathcal{G}$ is clear from the context) we denote the set of vertices that contains u, v and all vertices in between u and v on level i in $\mathcal{G}$. We say that two vertices u and v are *consecutive in $\mathcal{G}$* when $[u, v] = \{u, v\}$. Two edges $e = (u, w), e' = (v, x)$ are *consecutive in $\mathcal{G}$* when the only edges with one endpoint in $[u, v]_\mathcal{G}$ and the other endpoint in $[w, x]_\mathcal{G}$ are e and e'.

A *flow network* $F = (N, A)$ consists of a set of nodes N connected by a set of directed arcs A. Each arc has a *demand* specified by a function $d : A \to \mathbb{N}_0$ and a *capacity* specified by a function $c : A \to \mathbb{N} \cup \{\infty\}$ where ∞ encodes unlimited capacity. A *circulation* in F is a function $\varphi : A \to \mathbb{N}_0$ that assigns an integral flow to each arc of F and satisfies the two following conditions. First, the circulation has to respect the demands and capacities of the arcs, i.e., for each arc $a \in A$ it is $d(a) \leq \varphi(a) \leq c(a)$. Second, the circulation has to respect flow conservation, i.e., for each node $v \in N$ it is $\sum_{(u,v) \in A} \varphi(u, v) = \sum_{(v,u) \in A} \varphi(v, u)$. Depending on the flow network no circulation may exist.

3   Flow Model

In this section, we model the λ-DRAWABILITY as a problem of finding a circulation in a flow network. Let $\mathcal{G}$ be an embedded proper k-level graph together with a level-planar drawing Γ. As a first step, we add two directed paths p_left and p_right that consist of one vertex on each level from 1 to k to $\mathcal{G}$. Insert p_left and p_right into Γ to the left and right of all other vertices as the *left* and *right boundary*, respectively. See Fig. 2(a) and (c). From now on, we assume that $\mathcal{G}$ and Γ contain the left and right boundary.

The flow network $F_\mathcal{G}^\lambda$ consists of nodes and arcs and is similar to a directed dual of $\mathcal{G}$ with the difference that it takes the levels of $\mathcal{G}$ into account. In particular, for every edge e of $\mathcal{G}$, $F_\mathcal{G}^\lambda$ contains two nodes e_left and e_right, in the left and the right faces incident to e, and a dual *slope arc* $e^\star = (e_\text{right}, e_\text{left})$ with demand 0 and capacity $\lambda - 1$; see the blue arcs in Fig. 2(b) and (c). The flow across $e^\star$ determines the slope of e. Additionally, for every pair of consecutive vertices u, v we add two nodes $[u, v]_\text{low}$ and $[u, v]_\text{high}$ to $F_\mathcal{G}^\lambda$ and connect them by a *space arc* $[u, v]^\star$; see the red arcs in Fig. 2(b) and (c). The flow across $[u, v]^\star$ determines the space between u and v. The space between u and v needs to be
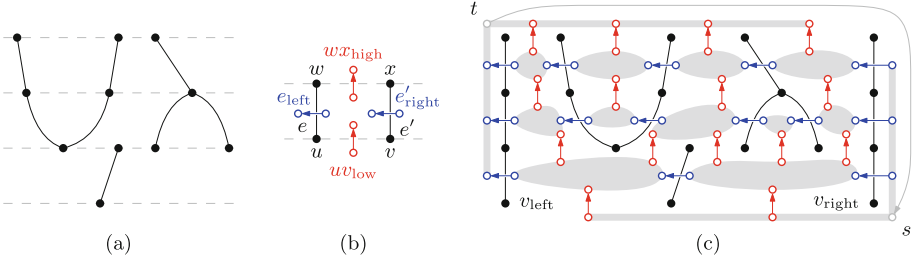
Fig. 2. (a) An embedded level graph $\mathcal{G}$. (b) The definition of the arcs of the flow network. (c) The graph $\mathcal{G}$ together with the paths p_{left} and p_{right} in black. The resulting flow network $F_{\mathcal{G}}^{\lambda}$ (c) consists of the blue slope arcs and the red space arcs, its nodes are formed by merging the nodes in the gray areas. The red space arcs have a demand of 1 and a capacity of $(\lambda - 1)(n - 1)$ and the blue slope arcs have a demand of zero and a capacity of $\lambda - 1$.

at least one to prevent u and v from colliding and can be at most $(\lambda - 1)(n - 1)$ due to the restriction to compact drawings. So, assign to $[u, v]^{\star}$ a demand of one and a capacity of $(\lambda - 1)(n - 1)$. To obtain the final flow network we merge certain nodes. Let $e = (u, w)$ and $e' = (v, x)$ be consecutive edges. Merge the nodes $e_{\text{right}}, e'_{\text{left}}$, the nodes $\{\{u', v'\}_{\text{high}} : \forall u', v' \text{ consecutive in } [u, v]\}$ and the nodes $\{\{w', x'\}_{\text{low}} : \forall w', x' \text{ consecutive in } [w, x]\}$ into a single node. Next, merge all remaining source and sink nodes into one source node s and one sink node t, respectively. See Fig. 2(c), where the gray areas touch nodes that are merged into a single node. Finally, insert an arc from t to s with unlimited capacity.

The network $F_{\mathcal{G}}^{\lambda}$ is designed in such a way that the circulations in $F_{\mathcal{G}}^{\lambda}$ correspond bijectively to the λ-drawings of $\mathcal{G}$. Let Γ be a drawing of $\mathcal{G}$ and let x be the function that assigns to each vertex of $\mathcal{G}$ its x-coordinate in Γ. We define a dual circulation $x^{\star}$ as follows. Recall that every arc a of $F_{\mathcal{G}}^{\lambda}$ is either dual to an edge of $\mathcal{G}$ or to two consecutive vertices in $\mathcal{G}$. Hence, the left and right incident faces f_{left} and f_{right} of a in $F_{\mathcal{G}}^{\lambda}$ contain vertices of $\mathcal{G}$. Define the circulation $x^{\star}$ by setting $x^{\star}(a) := x(f_{\text{right}}) - x(f_{\text{left}})$. We remark the following, although we defer the proof to the next section.

Lemma 1. *Let $\mathcal{G}$ be an embedded proper level-planar graph together with a λ-drawing Γ. The dual $x^{\star}$ of the function x that assigns to each vertex of $\mathcal{G}$ its x-coordinate in Γ is a circulation in $F_{\mathcal{G}}^{\lambda}$.*

In the reverse direction, given a circulation φ in $F_{\mathcal{G}}^{\lambda}$ we define a dual function $\varphi^{\star}$ that, when interpeted as assigning an x-coordinate to the vertices of $\mathcal{G}$, defines a λ-drawing of G. Refer to the level-1-vertex p_{right} as v_{right}. Start by setting $\varphi^{\star}(v_{\text{right}}) = 0$, i.e., the x-coordinate of v_{right} is 0. Process the remaining vertices of the right boundary in ascending order with respect to their levels. Let (u, v) be an edge of the right boundary so that u has already been processed and v has not been processed yet. Then set $\varphi^{\star}(v) = \varphi^{\star}(u) + \varphi((u, v)^{\star})$, where $(u, v)^{\star}$ is the slope arc dual to (u, v). Let w, x be a pair of consecutive

vertices so that x has already been processed and w has not yet been processed yet. Then set $\varphi^\star(w) = \varphi^\star(x) + \varphi([w,x]^\star)$, where $[w,x]^\star$ is a space arc. It turns out that $\varphi^\star$ defines a λ-drawing of $\mathcal{G}$.

Lemma 2. *Let $\mathcal{G}$ be an embedded proper level-planar graph, let $\lambda \in \mathbb{N}$ and let φ be a circulation in $F_\mathcal{G}^\lambda$. Then the dual $\varphi^\star$, when interpeted as assigning an x-coordinate to the vertices of $\mathcal{G}$, defines a λ-drawing of G.*

While both Lemmas 1 and 2 can be proven directly, we defer their proofs to Sect. 4 where we introduce the distance model and prove Lemmas 3 and 4, the stronger versions of Lemmas 1 and 2, respectively. Combining Lemmas 1 and 2 we obtain the following.

Theorem 1. *Let $\mathcal{G}$ be an embedded proper level-planar graph and let $\lambda \in \mathbb{N}$. The circulations in $F_\mathcal{G}^\lambda$ correspond bijectively to the λ-drawings of $\mathcal{G}$.*

Theorem 1 implies that a λ-drawing can be found by applying existing flow algorithms to $F_\mathcal{G}^\lambda$. For that, we transform our flow network with arc demands to the standard maximum flow setting without demands by introducing new sources and sinks. We can then use the $O(n \log^3 n)$-time multiple-source multiple-sink maximum flow algorithm due to Borradaile et al. [6] to find a circulation in $F_\mathcal{G}^\lambda$ or to determine that no circulation exists.

Corollary 1. *Let $\mathcal{G}$ be an embedded proper level-planar graph and let $\lambda \in \mathbb{N}$. It can be tested in $O(n \log^3 n)$ time whether a λ-drawing of $\mathcal{G}$ exists, and if so, such a drawing can be found within the same running time.*

3.1 Connected Partial Drawings

Recall that a partial λ-drawing is a tuple $(\mathcal{G}, \mathcal{H}, \Pi)$, where $\mathcal{G}$ is an embedded level-planar graph, $\mathcal{H}$ is an embedded subgraph of $\mathcal{G}$ and Π is a λ-drawing of $\mathcal{H}$. We say that $(\mathcal{G}, \mathcal{H}, \Pi)$ is λ-*extendable* if $\mathcal{G}$ admits a λ-drawing Γ whose restriction to $\mathcal{H}$ is Π. Here Γ is referred to as a λ-*extension* of $(\mathcal{G}, \mathcal{H}, \Pi)$.

In this section we show that in case $\mathcal{H}$ is connected, we can use the flow model to decide whether $(\mathcal{G}, \mathcal{H}, \Pi)$ is λ-extendable. Observe that when $\mathcal{H}$ is connected Π is completely defined by the slopes of the edges in $\mathcal{H}$ up to horizontal translation. Let $F_\mathcal{G}^\lambda$ be the flow network corresponding to $\mathcal{G}$. In order to fix the slopes of an edge e of $\mathcal{H}$ to a value ℓ, we fix the flow across the dual slope arc $e^\star$ in $\mathcal{H}$ to ℓ. Checking whether a circulation in the resulting flow network exists can be reduced to a multiple-source multiple-sink maximum flow problem, which once again can be solved by the algorithm due to Borradaile et al. [6].

Corollary 2. *Let $(\mathcal{G}, \mathcal{H}, \Pi)$ be a partial λ-drawing where $\mathcal{H}$ is connected. It can be tested in $O(n \log^3 n)$ time whether $(\mathcal{G}, \mathcal{H}, \Pi)$ is λ-extendable, and if so, a corresponding λ-extension can be constructed within the same running time.*

4   Dual Distance Model

A minimum cut (and, equivalently, the value of the maximum flow) of an st-planar graph G can be determined by computing a shortest $(s^\star, t^\star)$-path in a dual of G [17,18]. Hassin showed that to construct a flow, it is sufficient to compute the distances from $s^\star$ to all other vertices in the dual graph [13]. To the best of our knowledge, this duality has been exploited only for flow networks with arc capacities, but not with arc demands. In this section, we extend this duality to arcs with demands. The resulting dual distance model improves the running time for the λ-DRAWABILITY, lets us test the existence of λ-extensions of partial λ-drawings for non-connected subgraphs, and allows us to develop an efficient algorithm for testing the existence of simultaneous λ-drawings.

We define $D_{\mathcal{G}}^\lambda$ to be the directed dual of $F_{\mathcal{G}}^\lambda$ as follows. Let $a = (u, v)$ be an arc of $F_{\mathcal{G}}^\lambda$ with demand $d(a)$ and capacity $c(a)$. Further, let f_{left} and f_{right} denote the left and the right faces of a in $F_{\mathcal{G}}^\lambda$, respectively. The dual $D_{\mathcal{G}}^\lambda$ contains f_{left} and f_{right} as vertices connected by one edge $(f_{\text{left}}, f_{\text{right}})$ with length $c(a)$ and another edge $(f_{\text{right}}, f_{\text{left}})$ with length $-d(a)$; see Fig. 3.

Observe that to obtain $F_{\mathcal{G}}^\lambda$ from $\mathcal{G}$ (with left and right paths p_{left} and p_{right}) we added dual arcs to edges of $\mathcal{G}$ and dual arcs to the space between two consecutive vertices on one level. Consider for a moment the graph $\mathcal{G}'$ obtained from $\mathcal{G}$ by adding edges (u, v) for all consecutive vertices u v, where u is to the right of v. Graph G' and $D_{\mathcal{G}}^\lambda$ are identical and therefore $D_{\mathcal{G}}^\lambda$ has the vertex set V of $\mathcal{G}$ and contains a subset of its edges. Recall that the dual slope arcs in $F_{\mathcal{G}}^\lambda$ have demand 0 and capacity $\lambda - 1$, therefore the edges of $D_{\mathcal{G}}^\lambda$ that connect vertices on different layers have non-negative length. While the edges of $D_{\mathcal{G}}^\lambda$ between consecutive vertices on the same level have length -1.
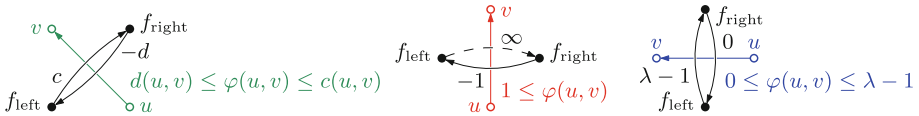

Fig. 3. Definition of the dual edges for a flow network arc $a = (u, v)$ with demand $d(a)$ and capacity $c(a)$. Let f_{left} and f_{right} denote the vertices corresponding to the faces to the left and right of a in $F_{\mathcal{G}}^\lambda$. Then add the edge $(f_{\text{left}}, f_{\text{right}})$ with length $c(a)$ and the reverse edge $(f_{\text{right}}, f_{\text{left}})$ with length $-d(a)$. Edges with infinite length are not created because they do not add constraints.

A *distance labeling* is a function $x : V \to \mathbb{Z}$ that for every edge (u, v) of $D_{\mathcal{G}}^\lambda$ with length l satisfies $x(v) \leq x(u) + l$. We also say that (u, v) *imposes the distance constraint* $x(v) \leq x(u) + l$. A distance labeling for $D_{\mathcal{G}}^\lambda$ is the x-coordinate assignment for a λ-drawing: For an edge (u, v) of $D_{\mathcal{G}}^\lambda$ where u, v are consecutive vertices in $\mathcal{G}$, the distance labeling guarantees $x(v) \leq x(u) - 1$, i.e., the consecutive vertices are in the correct order and do not overlap. If an edge (u, v) between layers has length $\lambda - 1$, then the distance labeling ensures $x(v) \leq x(u) + \lambda - 1$, i.e., (u, v)
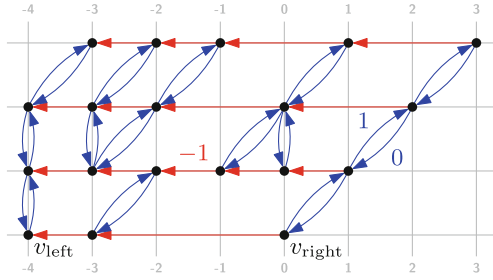
Fig. 4. The distance network $D_{\mathcal{G}}^2$ obtained from the flow network $F_{\mathcal{G}}^2$ shown in Fig. 2(c). The x-coordinate of every vertex is its distance from v_{right} in $D_{\mathcal{G}}^2$. All red arcs have length -1, all blue arcs pointing up have length 1 and all blue arcs pointing down have length 0. For every red arc there exists an arc in the reverse direction with length ∞. We omit these arcs because they do not impose any constraints on the shortest distance labeling.

has a slope in $\{0, \ldots, \lambda-1\}$. Computing the shortest distances from v_{right} in $D_{\mathcal{G}}^{\lambda}$ to every vertex (if they are well-defined) gives a distance labeling that we refer to as the *shortest distance labeling*. A distance labeling of $D_{\mathcal{G}}^{\lambda}$ does not necessarily exist. This is the case when $D_{\mathcal{G}}^{\lambda}$ contains a negative cycle, e.g., when the in- or out-degree of a vertex in $\mathcal{G}$ is strictly larger than λ. For a distance labeling x of $D_{\mathcal{G}}^{\lambda}$ we define a dual circulation $x^{\star}$ by setting $x^{\star}(a) := x(f_{\text{right}}) - x(f_{\text{left}})$ for each arc a of $F_{\mathcal{G}}^{\lambda}$ with left and right incident faces f_{left} and f_{right} (Fig. 4).

Lemma 3. *Let $\mathcal{G}$ be an embedded level-planar graph and Γ be a λ-drawing of $\mathcal{G}$. The function x that assigns to each vertex of $\mathcal{G}$ its x-coordinate in Γ is a distance labeling of $D_{\mathcal{G}}^{\lambda}$ and its dual $x^{\star}$ is a circulation in $F_{\mathcal{G}}^{\lambda}$.*

Proof. Since Γ preserves the embedding of $\mathcal{G}$, for each consecutive vertices v, u, with v preceeding u in $\mathcal{G}$ it holds that $\Gamma(v) < \Gamma(u)$. Since Γ is a grid drawing $\Gamma(v) \leq \Gamma(u) - 1$, which implies $x(v) \leq x(u) + \ell$, where $\ell = -1$ is the length of (u,v). Since Γ is a λ-drawing, i.e. every edge (u,v) between the two levels has a slope in $\{0, \ldots \lambda-1\}$, it holds that $\Gamma(u) \leq \Gamma(v) \leq \Gamma(u) + \lambda - 1$, which implies $x(u) \leq x(v) + 0$, for the edge (v,u) of $D_{\mathcal{G}}^{\lambda}$ with length zero and $x(v) < x(u) + \lambda - 1$ for the edge (u,v) of $D_{\mathcal{G}}^{\lambda}$ with length $\lambda - 1$. Hence, x is a distance labeling of $D_{\mathcal{G}}^{\lambda}$.

We now show that $x^{\star}$ is a circulation in $F_{\mathcal{G}}^{\lambda}$. Let $f_1, f_2, \ldots, f_t, f_{t+1} = f_1$ be the faces incident to some node v of $F_{\mathcal{G}}^{\lambda}$ in counter-clockwise order. Let a be the arc incident to v and dual to the edge between f_i and f_{i+1} with $1 \leq i \leq t$. If a is an incoming arc, it adds a flow of $x(f_{i+1}) - x(f_i)$ to v. If a is an outgoing arc, it removes a flow of $x(f_i) - x(f_{i+1})$ from v, or, equivalently, it adds a flow of $x(f_{i+1}) - x(f_i)$ to v. Therefore, the flow through v is $\sum_i (x(f_{i+1}) - x(f_i))$. This sum cancels to zero, i.e., the flow is preserved at v. Recall that the edge $(f_{\text{left}}, f_{\text{right}})$ with length $c(a)$ in $D_{\mathcal{G}}^{\lambda}$ ensures $x(f_{\text{right}}) \leq x(f_{\text{left}}) + c(a)$, which gives $x^{\star}(a) \leq c(a)$. So, no capacities are exceeded. Analogously, the edge $(f_{\text{right}}, f_{\text{left}})$ with length $-d(a)$ in $D_{\mathcal{G}}^{\lambda}$ ensures $x(f_{\text{left}}) \leq x(f_{\text{right}}) - d(a)$,

which gives $x^\star(a) \geq d(a)$. Hence, all demands are fulfilled and $x^\star$ is indeed a circulation in $F_\mathcal{G}^\lambda$.   □

Recall from Sect. 3 that for a circulation φ in $F_\mathcal{G}^\lambda$ we define a dual drawing $\varphi^\star$ by setting the x-coordinates of the vertices of $\mathcal{G}$ as follows. For the lowest vertex of the right boundary set $\varphi^\star(v_\mathrm{right}) = 0$. Process the remaining vertices of the right boundary in ascending order with respect to their levels. Let (u, v) be an edge of the right boundary so that u has already been processed and v has not been processed yet. Then set $\varphi^\star(v) = \varphi^\star(u) + \varphi((u, v)^\star)$, where $(u, v)^\star$ is the slope arc dual to (u, v). Let w, x be a pair of consecutive vertices so that x has already been processed and w has not yet been processed yet. Then set $\varphi^\star(w) = \varphi^\star(x) + \varphi([w, x]^\star)$, where $[w, x]^\star$ is a space arc. It turns out that $\varphi^\star$ is a distance labeling of $D_\mathcal{G}^\lambda$ and a λ-drawing of $\mathcal{G}$.

Lemma 4. *Let $\mathcal{G}$ be an embedded level-planar graph, let $\lambda \in \mathbb{N}$, and let φ be a circulation in $F_\mathcal{G}^\lambda$. The dual $\varphi^\star$ is a distance labeling of $D_\mathcal{G}^\lambda$ and the drawing induced by interpreting the distance label of a vertex as its x-coordinate is a λ-drawing of $\mathcal{G}$.*

Proof. We show that $\varphi^\star$ is a distance labeling in $D_\mathcal{G}^\lambda$. The algorithm described above assings a value to every vertex of $D_\mathcal{G}^\lambda$. We now show that $\varphi^\star$ is indeed a distance labeling by showing that every edge satisfies a distance constraint.

Observe that the distance constraints imposed by edges dual to the space arcs are satisfied by construction. To show that the distance constraints imposed by edges dual to the slope arcs are also satisfied, we prove that for every edge (u, v), it holds that $\varphi^\star(v) = \varphi^\star(u) + \varphi((u, v)^\star)$. We refer to this as *condition* $\mathcal{C}$ for short. Since $\varphi((u, v)^\star) \leq \lambda - 1$ and the length ℓ of (u, v) is $\lambda - 1$ we obtain $\varphi^\star(v) = \varphi^\star(u) + \ell$, which implied that $\phi^\star$ is a distance labeling of $D_\mathcal{G}^\lambda$.

The proof is by induction based on the bottom to top and right to left order among the edges of $D_\mathcal{G}^\lambda$. We say that (a, b) *precedes* (c, d) if either $\ell(a) < \ell(c)$, or $\ell(a) = \ell(c)$ and a is to the right of c, or $\ell(a) = \ell(c)$ and b is to the right of d (in case $a = c$). For the base case observe that the edges with both end-vertices on the first level and the edges of p_right satisfy condition $\mathcal{C}$ by the definition of $\varphi^\star$. Now let (u, v) be an edge not addressed in the base case and assume that for every edge (u', v') preceding edge (u, v) condition $\mathcal{C}$ holds. For the inductive step we show that condition $\mathcal{C}$ also holds for (u, v). Let (u', v') denote the edge to the right of (u, v) so that (u, v) and (u', v') are consecutive; see Fig. 5. Because v is not the rightmost vertex on its level this edge exists. Let A denote the set of space arcs $v_1 v_2^\star$ in $F_\mathcal{G}^\lambda$ with $v_1, v_2 \in [v', v]$. Analogously, let B denote the set of space arcs $u_1 u_2^\star$ in $F_\mathcal{G}^\lambda$ with $u_1, u_2 \in [u', u]$. It is $\varphi^\star(v) = \varphi^\star(v') + \sum_{a \in A} \varphi(a)$ by definition of $\varphi^\star$. Further, by induction hypothesis and since (u', v') precedes (u, v) it holds that $\varphi^\star(v') = \varphi^\star(u') + \varphi((u', v')^\star)$. Inserting the latter into the former equation, we obtain $\varphi^\star(v) = \varphi^\star(u') + \varphi((u', v')^\star) + \sum_{a \in A} \varphi(a)$. Again, by definition of $\varphi^\star$, it is $\varphi^\star(u) = \varphi^\star(u') + \sum_{b \in B} \varphi(b)$. By subtracting $\varphi^\star(u)$ from $\varphi^\star(v)$ we obtain

$$\varphi^\star(v) = \varphi^\star(u) - \sum_{b \in B} \varphi(b) + \varphi((u', v')^\star) + \sum_{a \in A} \varphi(a) \tag{1}$$

Flow conservation on the vertex of $F_{\mathcal{G}}^{\lambda}$ to which edges of A and B are incident gives $\varphi((u,v)^{\star}) - \sum_{a\in A}\varphi(a) - \varphi((u',v')^{\star}) + \sum_{b\in B}\varphi(b) = 0$. Solving this equation for $\varphi(u,v)$ and inserting it into (1) yields $\varphi^{\star}(v) = \varphi^{\star}(u) + \varphi((u,v)^{\star})$, i.e. the condition $\mathcal{C}$ holds for (u,v). Therefore $\varphi^{\star}$ is a distance labeling, which we have shown to define a λ-drawing of $\mathcal{G}$. □
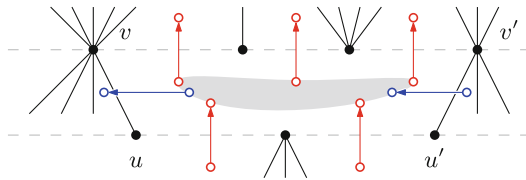

Fig. 5. Proof of Lemma 4. Sets A and B contain the outgoing and incoming red flow network arcs incident to the gray oval, respectively.

Because $D_{\mathcal{G}}^{\lambda}$ is planar we can use the $O(n\log^2 n/\log\log n)$-time shortest path algorithm due to Mozes and Wulff-Nilsen [26] to compute the shortest distance labeling. This improves our $O(n\log^3 n)$-time algorithm from Sect. 3.

Theorem 2. *Let $\mathcal{G}$ be an embedded proper level-planar graph. The distance label-ings of $D_{\mathcal{G}}^{k}$ correspond bijectively to the λ-drawings of $\mathcal{G}$. If such a drawing exists, it can be found in $O(n\log^2 n/\log\log n)$ time.*

5   Partial and Simultaneous Drawings

In this section we use the distance model from Sect. 4 to construct partial and simultaneous λ-drawings. We start with introducing a useful kind of drawing. Let Γ be a λ-drawing of $\mathcal{G}$. We call Γ a λ-*rightmost* drawing when there exists no λ-drawing Γ' with $\Gamma(v) < \Gamma'(v)$ for some $v \in V$. In this definition, we assume $x(\Gamma(v_{\text{right}})) = x(\Gamma'(v_{\text{right}})) = 0$ to exclude trivial horizontal translations. Hence, a drawing is rightmost when every vertex is at its rightmost position across all level-planar λ-slope grid drawings of $\mathcal{G}$. It is not trivial that a λ-rightmost drawing exists, but it follows directly from the definition that if such a drawing exists, it is unique. The following lemma establishes the relationship between λ-rightmost drawings and shortest distance labelings of $D_{\mathcal{G}}^{\lambda}$.

Lemma 5. *Let $\mathcal{G}$ be an embedded proper level-planar graph. If $D_{\mathcal{G}}^{\lambda}$ has a shortest distance labeling it describes the λ-rightmost drawing of $\mathcal{G}$.*

Proof. The shortest distance labeling of $D_{\mathcal{G}}^{\lambda}$ is maximal in the sense that for any vertex v there exists a vertex u and an edge (u,v) with length l so that it is $x(v) = x(u) + l$. Recall that the definition of distance labelings only requires $x(v) \leq x(u) + l$. The claim then follows by induction over V in ascending order with respect to the shortest distance labeling. □

5.1   Partial Drawings

Let $(\mathcal{G}, \mathcal{H}, \Pi)$ be a partial λ-drawing. In Sect. 3.1 we have shown that the flow model can be adapted to check whether $(\mathcal{G}, \mathcal{H}, \Pi)$ has a λ-extension, in case $\mathcal{H}$ is connected. In this section, we show how to adapt the distance model to extend partial λ-drawings, including the case $\mathcal{H}$ is disconnected. Recall that the distance label of a vertex v is its x-coordinate. A partial λ-drawing fixes the x-coordinates of the vertices of $\mathcal{H}$. The idea is to express this with additional constraints in $D_{\mathcal{G}}^{\lambda}$. Let v_{ref} be a vertex of $\mathcal{H}$. In a λ-extension of $(\mathcal{G}, \mathcal{H}, \Pi)$, the relative distance along the x-axis between a vertex v of $\mathcal{H}$ and vertex v_{ref} should be $d_v = \Pi(v_{\text{ref}}) - \Pi(v)$. This can be achieved by adding an edge (v, v_{ref}) with length d_v and an edge (v_{ref}, v) with length $-d_v$. The first edge ensures that it is $x(v_{\text{ref}}) \leq x(v) + d_v$, i.e., $x(v) \geq x(v_{\text{ref}}) - d_v$ and the second edge ensures $x(v) \leq x(v_{\text{ref}}) - d$. Together, this gives $x(v) = x(v_{\text{ref}}) - d_v$. Let $D_{\mathcal{G}, \Pi}^{\lambda}$ be $D_{\mathcal{G}}^{\lambda}$ augmented by the edges $\{(v, v_{\text{ref}}), (v_{\text{ref}}, v) : \forall v \in \mathcal{H}\}$ with lengths as described above.

To decide existence of λ-extension and in affirmative construct the corresponding drawing we compute the shortest distance labeling in $D_{\mathcal{G}, \Pi}^{\lambda}$. Observe that this network can contain negative cycles and therefore no shortest distance labeling. Unfortunately, $D_{\mathcal{G}, \Pi}^{\lambda}$ is not planar, and thus we cannot use the embedding-based algorithm of Mozes and Wulff-Nilsen. However, since all newly introduced edges have v_{ref} as one endpoint, v_{ref} is an *apex* of $D_{\mathcal{G}}^{\lambda}$, i.e., removing v_{ref} from $D_{\mathcal{G}, \Pi}^{\lambda}$ makes it planar. Therefore $D_{\mathcal{G}, \Pi}^{\lambda}$ can be recursively separated by separators of size $O(\sqrt{n})$. We can therefore use the shortest-path algorithm due to Henzinger et al. to compute the shortest distance labeling of $D_{\mathcal{G}, \Pi}^{\lambda}$ in $O(n^{4/3} \log n)$ time [15].

Theorem 3. *Let $(\mathcal{G}, \mathcal{H}, \Pi)$ be a partial λ-drawing. In $O(n^{4/3} \log n)$ time it can be determined whether $(\mathcal{G}, \mathcal{H}, \Pi)$ has a λ-extension and in the affirmative the corresponding drawing can be computed within the same running time.*

5.2   Simultaneous Drawings

In the simultaneous λ-drawing problem, we are given a tuple $(\mathcal{G}_1, \mathcal{G}_2)$ of two embedded level-planar graphs that share a common subgraph $\mathcal{G}_{1 \cap 2} = \mathcal{G}_1 \cap \mathcal{G}_2$. We assume w.l.o.g. that G_1 and G_2 share the same right boundary and that the embeddings of $\mathcal{G}_1$ and $\mathcal{G}_2$ coincide on $\mathcal{G}_{1 \cap 2}$. The task is to determine whether there exist λ-drawings Γ_1, Γ_2 of $\mathcal{G}_1, \mathcal{G}_2$, respectively, so that Γ_1 and Γ_2 coincide on the shared graph $\mathcal{G}_{1 \cap 2}$. The approach is the following. Start by computing the rightmost drawings of $\mathcal{G}_1$ and $\mathcal{G}_2$. Then, as long as these drawings do not coincide on $\mathcal{G}_{1 \cap 2}$ add necessary constraints to $D_{\mathcal{G}_1}^{\lambda}$ and $D_{\mathcal{G}_2}^{\lambda}$. This process terminates after a polynomial number of iterations, either by finding a simultaneous λ-drawing, or by determining that no such drawing exist.

Finding the necessary constraints works as follows. Suppose that Γ_1, Γ_2 are the rightmost drawings of $\mathcal{G}_1, \mathcal{G}_2$, respectively. Because both $\mathcal{G}_1$ and $\mathcal{G}_2$ have the same right boundary they both contain vertex v_{right}. We define the coordinates in the distance labelings of $D_{\mathcal{G}_1}^{\lambda}$ and $D_{\mathcal{G}_2}^{\lambda}$ in terms of this reference vertex.

Now suppose that for some vertex v of $\mathcal{G}_{1\cap 2}$ the x-coordinates in Γ_1 and Γ_2 differ, i.e., it is $\Gamma_1(v) \neq \Gamma_2(v)$. Assume $\Gamma_1(v) < \Gamma_2(v)$ without loss of generality. Because Γ_1 is a rightmost drawing, there exists no drawing of $\mathcal{G}_1$ where v has an x-coordinate greater than $\Gamma_1(v)$. In particular, there exist no simultaneous drawings where v has an x-coordinate greater than $\Gamma_1(v)$. Therefore, we must search for a simultaneous drawing where $\Gamma_2(v) \leq \Gamma_1(v)$. We can enforce this constraint by adding an edge (v_{right}, v) with length $\Gamma_1(v)$ into $D^\lambda_{\mathcal{G}_2}$. We then attempt to compute the drawing Γ_2 of $\mathcal{G}_2$ defined by the shortest distance labeling in $D^\lambda_{\mathcal{G}_2}$. This attempt produces one of two possible outcomes. The first possibility is that there now exists a negative cycle in $D^\lambda_{\mathcal{G}_2}$. This means that there exists no drawing Γ_2 of G_2 with $\Gamma_2(v) \leq \Gamma(v)$. Because Γ_1 is a rightmost drawing, this means that no simultaneous drawings of $\mathcal{G}_1$ and $\mathcal{G}_2$ exist. The algorithm then terminates and rejects this instance. The second possiblity is that we obtain a new drawing Γ_2. This drawing is rightmost among all drawings that satisfy the added constraint $\Gamma_2(v) \leq \Gamma_1(v)$. In this case there are again two possibilities. Either we have $\Gamma_1(v) = \Gamma_2(v)$ for each vertex v in $\mathcal{G}_{1\cap 2}$. In this case Γ_1 and Γ_2 are simultaneous drawings and the algorithm terminates. Otherwise there exists at least one vertex w in $\mathcal{G}_{1\cap 2}$ with $\Gamma_1(w) \neq \Gamma_2(w)$. We then repeat the procedure just described for adding a new constraint.

We repeat this procedure of adding other constraints. To bound the number of iterations, recall that we only consider compact drawings, i.e., drawings whose width is at most $(\lambda - 1)(n - 1)$. In each iteration the x-coordinate of at least one vertex is decreased by at least one. Therefore, each vertex is responsible for at most $(\lambda - 1)(n - 1)$ iterations. The total number of iterations is therefore bounded by $n(\lambda - 1)(n - 1) \in O(\lambda n^2)$.

Note that due to the added constraints $D^\lambda_{\mathcal{G}_1}$ and $D^\lambda_{\mathcal{G}_2}$ are generally not planar. We therefore apply the $O(n^{4/3} \log n)$-time shortest-path algorithm due to Henzinger et al. that relies not on planarity but on $O(\sqrt{n})$-sized separators to compute the shortest distance labellings. This gives the following.

Theorem 4. *Let $\mathcal{G}_1, \mathcal{G}_2$ be embedded level-planar graphs that share a common subgraph $\mathcal{G}_{1\cap 2}$. In $O(\lambda n^{10/3} \log n)$ time it can be determined whether $\mathcal{G}_1, \mathcal{G}_2$ admit simultaneous λ-drawings and if so, such drawings can be computed within the same running time.*

6 Conclusion

In this paper we studied λ-drawings, i.e., level-planar drawings with λ slopes. We model λ-drawings of proper level-planar graphs as integer flow networks. This lets us find λ-drawings and extend connected partial λ-drawings in $O(n \log^3 n)$ time. We extend the duality between integer flows in a primal graph and shortest distances in its dual to obtain a more powerful distance model. This distance model allows us to find λ-drawings in $O(n \log^2 n / \log \log n)$ time, extend not-necessarily-connected partial λ-drawings in $O(n^{4/3} \log n)$ time and find simultaneous λ-drawings in $O(\lambda n^{10/3} \log n)$ time.

In the non proper case, testing the existence of a 2-drawing becomes NP-hard, even for biconnected graphs with maximum edge length two [7].

References

1. FAUSTEDITION. http://www.faustedition.net/macrogenesis/dag
2. Angelini, P., Chaplick, S., Cornelsen, S., Da Lozzo, G., Di Battista, G., Eades, P., Kindermann, P., Kratochvíl, J., Lipp, F., Rutter, I.: Simultaneous orthogonal planarity. In: Hu, Y., Nöllenburg, M. (eds.) GD 2016. LNCS, vol. 9801, pp. 532–545. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-50106-2_41
3. Angelini, P., Da Lozzo, G., Di Battista, G., Frati, F., Patrignani, M., Rutter, I.: Beyond level planarity. In: Hu, Y., Nöllenburg, M. (eds.) GD 2016. LNCS, vol. 9801, pp. 482–495. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-50106-2_37
4. Barát, J., Matouvsek, J., Wood, D.R.: Bounded-degree graphs have arbitrarily large geometric thickness. Electr. J. Comb. **13**(1), 3 (2006)
5. Bläsius, T., Kobourov, S.G., Rutter, I.: Simultaneous embedding of planar graphs. In: Tamassia [32], pp. 349–381
6. Borradaile, G., Klein, P.N., Mozes, S., Nussbaum, Y., Wulff-Nilsen, C.: Multiple-source multiple-sink maximum flow in directed planar graphs in near-linear time. In: Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science, pp. 170–179. IEEE Press, New York (2011)
7. Brückner, G., Krisam, N.D., Mchedlidze, T.: Level-planar drawings with few slopes. CoRR, abs/1907.13558v1 (2019)
8. Brückner, G., Rutter, I.: Partial and constrained level planarity. In: Klein, P.N. (ed.) Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 2000–2011. SIAM (2017)
9. Di Giacomo, E., Liotta, G., Montecchiani, F.: Drawing outer 1-planar graphs with few slopes. J. Graph Algorithms Appl. **19**(2), 707–741 (2015)
10. Dujmović, V., Eppstein, D., Suderman, M., Wood, D.R.: Drawings of planar graphs with few slopes and segments. Comput. Geom. **38**(3), 194–212 (2007)
11. Dujmović, V., Suderman, M., Wood, D.R.: Graph drawings with few slopes. Comput. Geom. **38**(3), 181–193 (2007)
12. Di Giacomo, E., Liotta, G., Montecchiani, F.: Drawing subcubic planar graphs with four slopes and optimal angular resolution. Theor. Comput. Sci. **714**, 51–73 (2018)
13. Hassin, R.: Maximum flow in (s,t) planar networks. Inform. Process. Lett. **13**(3), 107 (1981)
14. Healy, P., Nikolov, N.S.: Hierarchical drawing algorithms. In: Tamassia [32], pp. 409–453
15. Henzinger, M.R., Klein, P.N., Rao, S., Subramanian, S.: Faster shortest-path algorithms for planar graphs. J. Comput. Syst. Sci. **55**(1), 3–23 (1997)
16. Hoffmann, U.: On the complexity of the planar slope number problem. J. Graph Algorithms Appl. **21**(2), 183–193 (2017)
17. Te Chiang, H.: Integer Programming and Network Flows. Addison-Wesley, Reading (1969)
18. Itai, A., Shiloach, Y.: Maximum flow in planar networks. SIAM J. Comput. **8**(2), 135–150 (1979)

19. Jänicke, S., Geßner, A., Franzini, G., Terras, M., Mahony, S., Scheuermann, G.: Traviz: a visualization for variant graphs. DSH **30**(Suppl–1), i83–i99 (2015)
20. Keszegh, B., Pach, J., Pálvölgyi, D.: Drawing planar graphs of bounded degree with few slopes. SIAM J. Discrete Math. **27**(2), 1171–1183 (2013)
21. Keszegh, B., Pach, J., Pálvölgyi, D., Tóth, G.: Drawing cubic graphs with at most five slopes. Comput. Geom. **40**(2), 138–147 (2008)
22. Kindermann, P., Meulemans, W., Schulz, A.: Experimental analysis of the accessibility of drawings with few segments. J. Graph Algorithms Appl. **22**(3), 501–518 (2018)
23. Knauer, K.B., Micek, P., Walczak, B.: Outerplanar graph drawings with few slopes. Comput. Geom. **47**(5), 614–624 (2014)
24. Lenhart, W., Liotta, G., Mondal, D., Nishat, R.I.: Planar and plane slope number of partial 2-trees. In: Wismath, S., Wolff, A. (eds.) GD 2013. LNCS, vol. 8242, pp. 412–423. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-03841-4_36
25. Mchedlidze, T., Nöllenburg, M., Rutter, I.: Extending convex partial drawings of graphs. Algorithmica **76**(1), 47–67 (2016)
26. Mozes, S., Wulff-Nilsen, C.: Shortest paths in planar graphs with real lengths in $O(n\log^2 n/\log\log n)$ time. In: de Berg, M., Meyer, U. (eds.) ESA 2010. LNCS, vol. 6347, pp. 206–217. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15781-3_18
27. Nickel, S., Nöllenburg, M.: Drawing k-linear metro maps. CoRR, abs/1904.03039 (2019)
28. Nöllenburg, M.: A survey on automated metro map layout methods. In: Schematic Mapping Workshop. Essex, UK, April 2014
29. Pach, J., Pálvölgyi, D.: Bounded-degree graphs can have arbitrarily large slope numbers. Electr. J. Comb. **13**(1), N1 (2006)
30. Patrignani, M.: On extending a partial straight-line drawing. Int. J. Found. Comput. Sci. **17**(5), 1061–1070 (2006)
31. Purchase, H.C.: Metrics for graph drawing aesthetics. J. Vis. Lang. Comput. **13**(5), 501–516 (2002)
32. Tamassia, R. (ed.): Handbook on Graph Drawing and Visualization. Chapman and Hall/CRC, London/Boca Raton (2013)