Algorithmica (2022) 84:176–196
https://doi.org/10.1007/s00453-021-00884-x

Level-Planar Drawings with Few Slopes

Guido Brückner[1] · Nadine Krisam[1] · Tamara Mchedlidze[2]

Received: 24 December 2019 / Accepted: 12 October 2021 / Published online: 19 November 2021

Abstract
We introduce and study level-planar straight-line drawings with a fixed number λ of slopes. For proper level graphs (all edges connect vertices of adjacent levels), 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.

Keywords Graph drawing · Level planarity · Constrained drawings

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 [26], phylogenetic 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.

An extended abstract of this paper appeared in the proceeding of the International Symposium on Graph Drawing and Network Visualization 2019 [9]. The present extended version contains the proof of NP-hardness of the considered problem, which has not appeared in the conference version.

✉ Guido Brückner
brueckner@kit.edu

Tamara Mchedlidze
t.mtsentlintze@uu.nl

1 Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

2 Utrecht University, Utrecht, Netherlands

By far the most popular way to achieve this is to apply the *Sugiyama 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 [21].

The Sugiyama framework consists of several steps: elimination of directed cycles in the initial graph, assignment of vertices to layers, vertex ordering and coordinate assignment. During each of these steps several criteria are optimized, by leading to more readable visualizations, see e.g. [21]. 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 $\ell : V \to \{1, 2, \dots, k\}$ that assigns each vertex of G to one of k levels. We refer to G together with ℓ as to a *k-level graph* (or *level graph* for short). 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. 1a 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. In particular, the levels are equispaced apart. 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-line segments. 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 Sugiyama 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 steepness of the edges [21]. 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 it is $\ell(v) - \ell(u) = 1$, the slope of (u, v) is then simply $\Gamma(v) - \Gamma(u)$. 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 $\{0, 1, \dots, \lambda - 1\}$.

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.

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

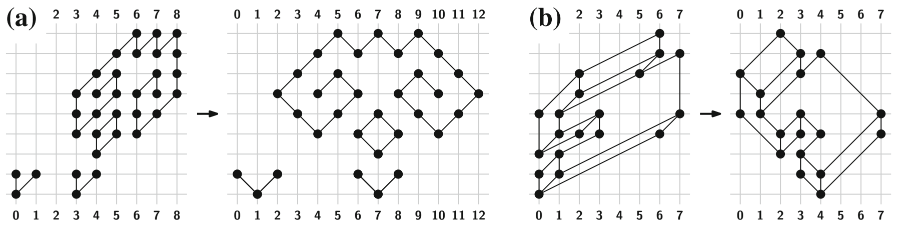

Fig. 1 Shearing drawings to change the slopes. In (a), the left drawing with slopes 0 and 1 is transformed into the right orthogonal drawing, i.e., one with slopes -1 and 1. In (b), the left drawing with slopes 0, 1 and 2 is transformed into a drawing with slopes -1, 0 and 1

been proposed as a measure of aesthetic quality of a graph drawing [40]. 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 [29]. 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 design principle often guides the visualization of metro maps. See [37] for a survey of the existing approaches, most of which generate octilinear layouts of metro maps, and [36] 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 [4,13,14,16,17,27,28,31,32,38]. Determining the planar slope number is hard in the existential theory of reals [23]. The slope number has also been studied for *upward* planar drawings, that is, drawings of directed graphs where each edge is drawn as a y-monotone curve (but, unlike with level planarity, the y-coordinate of the vertices is not prescribed) [5,15].

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 [33,39] and also the level-planar setting [10]. 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 [6] 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 complement these algorithmic results in Sect. 6 with a proof that 2-Drawability is NP-hard even for biconnected graphs where all edges have length one or two, and then finish with some concluding remarks in Sect. 7.

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 $(u, v) \in E$. For example, 2 is a gap in the left drawing in Fig. 1a. A drawing Γ is *compact* if it has no gap, e.g., the left drawing in Fig. 1b. Note that a λ-drawing of a connected level-planar graph is inherently compact. In a λ-drawing of a non-connected level-planar graph every gap can be eliminated by a horizontal shift. The fact that we only need to consider compact λ-drawings helps us to limit the drawing width. In particular, a compact λ-drawing has width 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'. For example, in Fig. 2b, the vertices u and v are consecutive, the vertices x and w are consecutive, and the edges (u, w) and (v, x) are consecutive; and in Fig. 2c, the vertices v_{left} and v_{right} are not consecutive.

A *flow network* $F = (N, A)$ consists of a set of nodes N connected by a set of directed arcs A. A node is a *source* if it has no incoming arcs and it is a *sink* if it has no outgoing arcs. A flow network is an *st-graph* if it has exactly one source and exactly one sink. 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 ∞ means 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 satisifies 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.

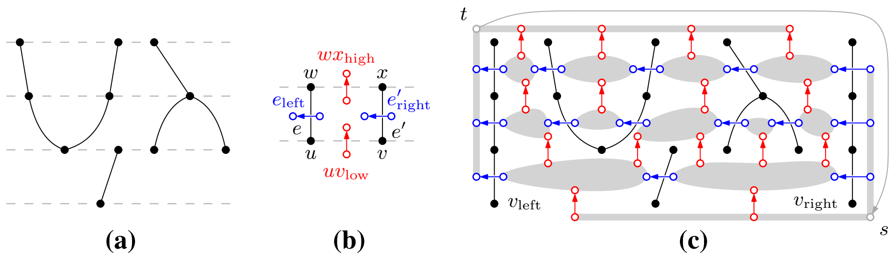

Fig. 2 An embedded level graph $\mathcal{G}$ (a). The definition of the arcs of the flow network (b). The graph $\mathcal{G}$ together with the paths p_{left} and p_{right} in black (c). The resulting flow network $F^{\lambda}_{\mathcal{G}}$ consists of the blue slope arcs and the red space arcs, its 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 $\lambda - 1$ (Color figure online)

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. 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 $\mathcal{G}$ to the left and right of all other vertices as the *left* and *right boundary*, respectively. See Fig. 2a and c.

From now on, we assume that $\mathcal{G}$ contains the left and right boundary.

The flow network $F^{\lambda}_{\mathcal{G}}$ 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 $\mathcal{G}$, $F^{\lambda}_{\mathcal{G}}$ 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. 2b 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 $F^{\lambda}_{\mathcal{G}}$ and connect them by a *space arc* $[u, v]^{\star}$; see the red arcs in Fig. 2b and c. The flow across $[u, v]^{\star}$ determines the space between u and v. The space between u and v needs to be 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_{right}, e'_{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. 2c, where the gray areas touch nodes that are merged into a single node. Observe that flow network is a connected st-graph. Clearly s is a source and t is a sink. Each remaining node v corresponds to two consecutive edges of $\mathcal{G}$, so by construction it has exactly one incoming and one outgoing slope arc, so v is neither a source nor a sink. This also implies that there exists a directed path of slope arcs from s to v, and a directed path of slope arcs from v to t, so the flow network is connected. Finally, to admit non-trivial circulations, insert an arc from t to s with capacity ∞. Observe that $F^{\lambda}_{\mathcal{G}}$ is planar by its construction based on the planar embedding of $\mathcal{G}$.

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 each arc a of $F_{\mathcal{G}}^{\lambda} - (s, t)$ is a slope arc or a space arc. If a is a slope arc it is dual to an edge (u, w) of $\mathcal{G}$. Then define $x^{\star}(a) := x(w) - x(u)$. If a is a space arc it is dual to consecutive vertices u, v of $\mathcal{G}$, where u appears left of v. Then define $x^{\star}(a) := x(v) - x(u)$. 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 interpreted as assigning an x-coordinate to the vertices of $\mathcal{G}$, defines a λ-drawing of $\mathcal{G}$. Refer to the level-1-vertex of 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 interpreted as assigning an x-coordinate to the vertices of $\mathcal{G}$, defines a λ-drawing of $\mathcal{G}$.*

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, first transform our flow network with arc demands to the standard single-source single-sink maximum flow setting without arc demands using the construction due to Kleinberg and Tardos [30, Chapter 7.7]. This construction adds one new "super-source" s^* and one new "super-sink" t^* to $F_{\mathcal{G}}^{\lambda}$, and connects them with arcs to the other nodes in $F_{\mathcal{G}}^{\lambda}$. In particular, the size of the resulting flow network is linear in the size of $F_{\mathcal{G}}^{\lambda}$. Subdivide the arcs incident to s^*, t^* and then remove s^*, t^* from the flow network, obtaining an instance of the multiple-source multiple-sink maximum flow problem. Note that this network is planar. Use the $O(n \log^3 n)$-time multiple-source multiple-sink maximum flow algorithm due to Borradaile et al. [8] 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 $F^{\lambda}_{\mathcal{G}}$ 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. [8].

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 [24,25]. 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 [20]. 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 $D^{\lambda}_{\mathcal{G}}$ to be the weighted directed dual of $F^{\lambda}_{\mathcal{G}}$ as follows. Let a be an arc of $F^{\lambda}_{\mathcal{G}}$ 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 $F^{\lambda}_{\mathcal{G}}$, respectively. The dual $D^{\lambda}_{\mathcal{G}}$ contains f_{left} and f_{right} as vertices connected by the directed edge $(f_{\text{left}}, f_{\text{right}})$ with weight $c(a)$ and the directed edge $(f_{\text{right}}, f_{\text{left}})$ with weight $-d(a)$; see Fig. 3. Equivalently, $D^{\lambda}_{\mathcal{G}}$ is obtained directly from $\mathcal{G}$ as follows. Recall that if a is a slope arc, it is $d(a) = 0, c(a) = \lambda - 1$ and $f_{\text{left}}, f_{\text{right}}$ correspond to vertices u, w connected by the edge (u, w) in G. So, create for each directed edge (u, w) of $\mathcal{G}$ the weighted directed edges (u, w) with weight $\lambda - 1$ and (w, u) with weight 0 in $D^{\lambda}_{\mathcal{G}}$; see Fig. 3a. If a is not a slope arc then a is a space arc and it is $d(a) = 1, c(a) = (\lambda - 1)(n - 1)$ and $f_{\text{left}}, f_{\text{right}}$ correspond to

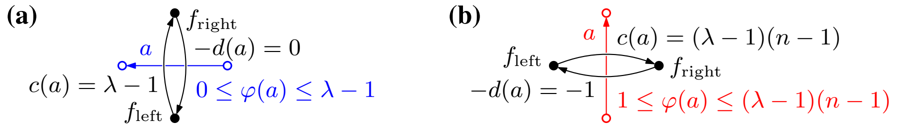

Fig. 3 Definition of the dual edges for a flow network arc $a = (u, v)$ with demand $d(a)$ and capacity $c(a)$. Let f_{left} and f_{right} denote the vertices corresponding to the faces to the left and right of a in $F^{\lambda}_{\mathcal{G}}$. Then add the edge $(f_{\text{left}}, f_{\text{right}})$ with weight $c(a)$ and the reverse edge $(f_{\text{right}}, f_{\text{left}})$ with weight $-d(a)$. The edge weights depend on whether a is a slope arc (a) or a space arc (b)

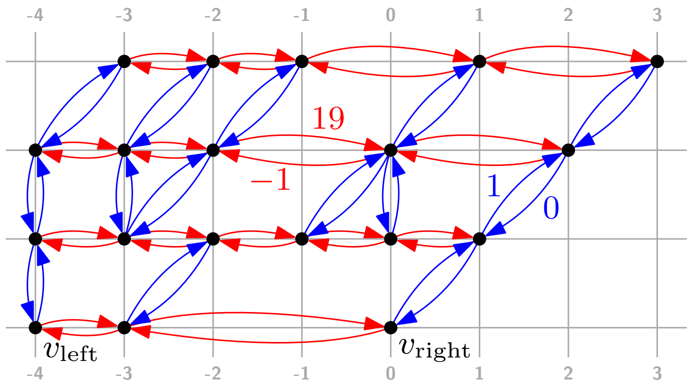

Fig. 4 The distance network $D^2_{\mathcal{G}}$ obtained from the flow network $F^2_{\mathcal{G}}$ shown in Fig. 2c. The x-coordinate of every vertex is its distance from v_{right} in $D^2_{\mathcal{G}}$. Red arcs pointing right have weight $(\lambda - 1)(n - 1) = 19$, red arcs pointing left have weight -1. Blue arcs pointing up have weight $\lambda - 1 = 1$ and blue arcs pointing down have weight 0 (Color figure online)

consecutive vertices u, v in $\mathcal{G}$, where u appears to the left of v. So, create for each pair of consecutive vertices u, v in $\mathcal{G}$ where u appears to the left of v the weighted directed edges (u, v) with weight $(\lambda - 1)(n - 1)$ and (v, u) with weight -1 in $D^{\lambda}_{\mathcal{G}}$; see Fig. 3b. Observe that $D^{\lambda}_{\mathcal{G}}$ has the vertex set V of $\mathcal{G}$ and a superset of its edges. Figure 4 shows the distance network obtained from the flow network shown in Fig. 2c.

A *distance labeling* is a function $x : V \to \mathbb{Z}$ that for every edge (u, v) of $D^{\lambda}_{\mathcal{G}}$ with weight 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 $D^{\lambda}_{\mathcal{G}}$ is the x-coordinate assignment for a λ-drawing: For an edge (u, v) of $D^{\lambda}_{\mathcal{G}}$ 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 weight $\lambda - 1$, then the distance labeling ensures $x(v) \leq x(u) + \lambda - 1$, i.e., (u, v) has a slope in $\{0, \ldots, \lambda - 1\}$. Computing the shortest distances from v_{right} in $D^{\lambda}_{\mathcal{G}}$ 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 $D^{\lambda}_{\mathcal{G}}$ does not necessarily exist. This is the case when $D^{\lambda}_{\mathcal{G}}$ 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 $D^{\lambda}_{\mathcal{G}}$ we define a dual circulation $x^{\star}$ as follows. Recall that each arc a of $F^{\lambda}_{\mathcal{G}} - (t, s)$ is a slope arc or a space arc. If a is a slope arc it is dual to an edge (u, w) of $\mathcal{G}$. Recall that u, w are vertices of $D^\lambda_\mathcal{G}$ and define $x^\star(a) := x(w) - x(u)$. If a is a space arc it is dual to consecutive vertices u, v of $\mathcal{G}$. Again, u, v are vertices of $D^\lambda_\mathcal{G}$, define $x^\star(a) := x(v) - x(u)$.

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 $D^\lambda_\mathcal{G}$ and its dual $x^\star$ is a circulation in $F^\lambda_\mathcal{G}$.*

Proof Since Γ preserves the embedding of $\mathcal{G}$, for each consecutive vertices $v,\ u$, with v preceding u in $\mathcal{G}$ it holds that $\Gamma(v) < \Gamma(u)$. Because Γ is a grid drawing $\Gamma(v) \leq \Gamma(u) - 1$, which implies $x(v) \leq x(u) + l$, where $l = -1$ is the weight 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 $D^\lambda_\mathcal{G}$ with weight zero and $x(v) \leq x(u) + \lambda - 1$ for the edge (u, v) of $D^\lambda_\mathcal{G}$ with weight $\lambda - 1$. Hence, x is a distance labeling of $D^\lambda_\mathcal{G}$.

We now show that $x^\star$ is a circulation in $F^\lambda_\mathcal{G}$. Let $f_1, f_2, \ldots, f_t, f_{t+1} = f_1$ be the faces incident to some node v of $F^\lambda_\mathcal{G}$ 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.

Now consider an arc a of $F^\lambda_\mathcal{G} - (t, s)$.

If a is a slope arc it is dual to an edge (u, w) of $\mathcal{G}$. Then $D^\lambda_\mathcal{G}$ contains the edge (u, w) with weight $c(a)$, which ensures $x(w) \leq x(u) + c(a)$, so $x^\star(a) \leq c(a)$. It also contains the edge (w, u) with weight $-d(a)$, which ensures $x(u) \leq x(w) - d(a)$, so $x^\star(a) \geq d(a)$.

If a is a space arc it is dual to consecutive vertices u, v of $\mathcal{G}$. Then $D^\lambda_\mathcal{G}$ contains the edge (u, v) with weight $c(a)$, which ensures $x(v) \leq x(u) + c(a)$, so $x^\star(a) \leq c(a)$. It also contains the edge (v, u) with weight $-d(a)$, which ensures $x(u) \leq x(v) - d(a)$, so $x^\star(a) \geq d(a)$.

Finally, the arc (t, s) has demand 0 and capacity ∞, so its demand is satisfied and its capacity is not exceeded.

Hence, $x^\star$ is indeed a circulation in $F^\lambda_\mathcal{G}$. □

Recall from Sect. 3 that for a circulation φ in $F^\lambda_\mathcal{G}$ 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 $\varphi^\star(v_{\text{right}}) = 0$. 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 $D^\lambda_\mathcal{G}$ and a λ-drawing of $\mathcal{G}$.

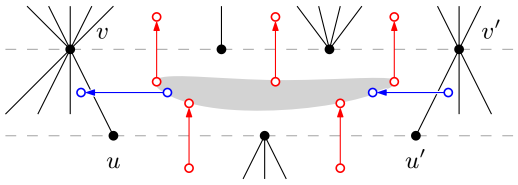

Fig. 5 Proof of Lemma 4. Sets A and B contain the incoming and outgoing red flow network arcs incident to the gray oval, respectively (Color figure online)

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, when interpreted as assigning an x-coordinate to the vertices of $\mathcal{G}$, defines 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 assigns 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 weight l of (u, v) is $l = \lambda - 1$ we obtain $\varphi^{\star}(v) = \varphi^{\star}(u) + \ell$, which implies that $\varphi^{\star}$ 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 $[u_1, u_2]^{\star}$ in $F_{\mathcal{G}}^{\lambda}$ with $u_1, u_2 \in [u', u]$. Analogously, let B denote the set of space arcs $[v_1, v_2]^{\star}$ in $F_{\mathcal{G}}^{\lambda}$ with $v_1, v_2 \in [v', v]$. It is $\varphi^{\star}(v) = \varphi^{\star}(v') - \sum_{b \in B} \varphi(b)$ by definition of $\varphi^{\star}$. Further, by induction hypothesis and since (u', v') precedes (u, v) it is $\varphi^{\star}(v') = \varphi^{\star}(u') + \varphi((u', v')^{\star})$. Inserting the latter into the former equation, we obtain

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

Again, by definition of $\varphi^{\star}$, it is $\varphi^{\star}(u) = \varphi^{\star}(u') - \sum_{a \in A} \varphi(a)$. Solving for $\varphi^{\star}(u')$ and inserting into (1) we obtain

$$\varphi^{\star}(v) = \varphi^{\star}(u) + \sum_{a \in A} \varphi(a) + \varphi((u', v')^{\star}) - \sum_{b \in B} \varphi(b). \tag{2}$$

Flow conservation on the vertex of $F_{\mathcal{G}}^{\lambda}$ to which edges of A and B are incident gives

$$\varphi((u', v')^{\star}) + \sum_{a \in A} \varphi(a) = \varphi((u, v)^{\star}) + \sum_{b \in B} \varphi(b). \tag{3}$$

Solving Eq. (3) for $\varphi((u', v')^{\star})$ and then inserting it into Eq. (2) yields $\varphi^{\star}(v) = \varphi^{\star}(u) + \varphi((u, v)^{\star})$, i.e., condition $\mathcal{C}$ holds for (u, v). Therefore $\varphi^{\star}$ is a distance labeling, which we have shown to define a λ-drawing of $\mathcal{G}$. □

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 [35] 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 labelings 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 weight 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 $D^{\lambda}_{\mathcal{G}}$. 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 weight d_v and an edge (v_{ref}, v) with weight $-d_v$. The first edge ensures that it is $x(v_{\text{ref}}) \le x(v) + d_v$, i.e., $x(v) \ge x(v_{\text{ref}}) - d_v$ and the second edge ensures $x(v) \le x(v_{\text{ref}}) - d$. Together, this gives $x(v) = x(v_{\text{ref}}) - d_v$. Let $D^{\lambda}_{\mathcal{G},\Pi}$ be $D^{\lambda}_{\mathcal{G}}$ augmented by the edges $\{(v, v_{\text{ref}}), (v_{\text{ref},v}) : \forall v \in \mathcal{H}\}$ with weights as described above.

To decide existence of λ-extension and in affirmative construct the corresponding drawing we compute the shortest distance labeling in $D^{\lambda}_{\mathcal{G},\Pi}$. Observe that this network can contain negative cycles and therefore no shortest distance labeling. Unfortunately, $D^{\lambda}_{\mathcal{G},\Pi}$ 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 $D^{\lambda}_{\mathcal{G}}$, i.e., removing v_{ref} from $D^{\lambda}_{\mathcal{G},\Pi}$ makes it planar. Therefore $D^{\lambda}_{\mathcal{G},\Pi}$ can be recursively separated by separators of size $O(\sqrt{n})$. The $O(n^{4/3} \log n)$-time shortest-path algorithm by Henzinger et al. [22] relies not on planarity but only on $O(\sqrt{n})$-sized separators [19, page 869]. So, run this algorithm to compute the shortest distance labeling of $D^{\lambda}_{\mathcal{G},\Pi}$.

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 λ-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 $D^{\lambda}_{\mathcal{G}_1}$ and $D^{\lambda}_{\mathcal{G}_2}$. 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 λ-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 $D^{\lambda}_{\mathcal{G}_1}$ and $D^{\lambda}_{\mathcal{G}_2}$ 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 $\Gamma_1(v) \ne \Gamma_2(v)$. 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 $\Gamma_2(v) \le \Gamma_1(v)$. We can enforce this constraint by adding an edge (v_{right}, v)

with weight $\Gamma_1(v)$ 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 possibility 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. However, all newly inserted arcs are incident to v_{right}, so v_{right} is an apex of $D^{\lambda}_{\mathcal{G}_i}$ for $i = 1, 2$. As in the previous section, we apply the $O(n^{4/3} \log n)$-time shortest-path algorithm by Henzinger et al. to compute the shortest distance labelings. 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 Complexity of the General Case

So far, we have considered λ-DRAWABILITY problem for proper level graphs, i.e., level graphs where all edges have length one. In this section, we consider the general case, where edges may have arbitrary lengths. We say that an edge with length two or more is *long*. One approach would be to try to adapt the flow model from Sect. 3 to this more general case. By subdividing long edges, any level graph G can be transformed into a proper level graph G'. Observe that two edges in G' created by subdividing the same long edge must have the same slope in order to yield a fixed-slope drawing of G. In the context of our flow model, this means that the amount of flow across the corresponding slope arcs must be the same. Our problem then becomes an instance of the *integer equal flow problem*. In this problem, we are given a flow network along with disjoint sets $R_1, R_2, \ldots, R_t$ of arcs. The task is to find the maximum flow from s to t such that the amount of flow across arcs in the same set R_i is the same. This problem was introduced and shown to be NP-hard by Sahni [41]. The problem remains NP-hard in special cases [18,42] and the integrality gap of the fractional LP can be arbitrarily large [34].

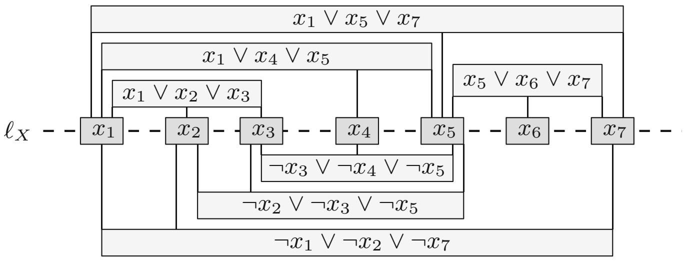

Fig. 6 An instance of planar monotone 3-SAT

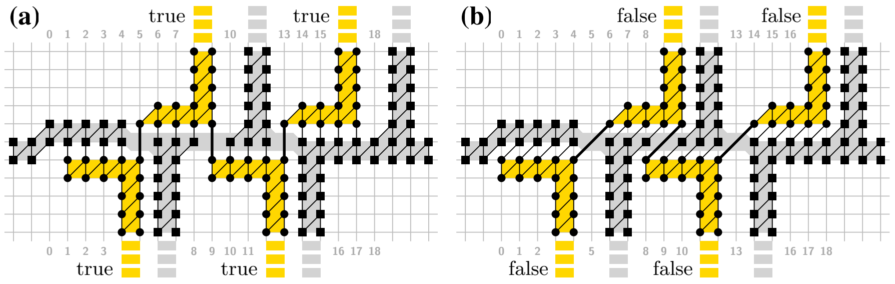

Fig. 7 The variable gadget drawn in the "true" configuration (a) and the "false" configuration (b)

In this section we show that λ-DRAWABILITY NP-complete even for $\lambda = 2$, biconnected graphs where all edges have length one or two. To this end, we present a reduction from *rectilinear planar monotone 3*-SAT [12]. An instance of this problem consists of a set of variables X and a set of clauses C. A clause is *positive* (*negative*) when it consists of only positive (negative) literals. We say that the instance is *monotone* when each clause is either positive or negative. The corresponding *variable-clause graph* consists of the vertices $X \cup C$ and each undirected edge $\{x, c\}$ where $x \in X$ is a variable that appears in clause $c \in C$. The variable-clause graph admits a planar drawing such that (i) the variables are aligned along a virtual horizontal line ℓ_X, (ii) positive clauses are drawn as vertices above ℓ_X, and symmetrically (iii)negative clauses are drawn as vertices below ℓ_X. See Fig. 6.

Our reduction works by first replacing every vertex that corresponds to a variable by a variable gadget and every vertex that corresponds to a positive (negative) clause by a positive (negative) clause gadget. All three gadgets consist of *fixed* and *movable* parts. The fixed parts only admit one level-planar two-slope drawing, whereas the movable parts admit two or more drawings depending on the choice of slope for some edges. Second, the gadgets are connected by a common fixed frame. All fixed parts of the gadgets are connected to the common frame in order to provide a common point of reference. The movable parts of the gadgets then interact in such a way that any level-planar two-slope drawing induces a solution to the underlying planar monotone 3-SAT instance.

The variable gadget consists of a number of *connectors* arranged around a fixed horizontal line that connects all variable gadgets along the virtual line of variables ℓ_X. See Fig. 7, where the fixed structure is shaded in gray. Vertices drawn as squares are fixed, i.e., they cannot change their position relative to other vertices drawn as squares. Vertices drawn as circles are movable, i.e., they can change their position relative to vertices drawn as squares. The line of variables ℓ_X extends from the the square vertices on the left and right boundaries of the drawing. Every connector consists of two *pins*: the movable *assignment pin* and the fixed *reference pin*. The variable gadget in Fig. 7 features four connectors: two above the horizontal line and two below the horizontal line. Assignment pins are shaded in yellow and reference pins are shaded in gray. The relative position of the assignment pin and the reference pin of one connector encodes the truth assignment of the underlying variable. Moreover, the reference pin allows the fixed parts of the clause gadgets to be connected to the variable gadgets and thereby to the common frame. Comparing Fig. 7a and b, observe how the relative position of the two pins of each connector changes depending on the truth assignment of the underlying variable. The key structure of the variable gadget is that the position of the assignment pins of one variable gadget are coordinated by long edges. In Fig. 7, long edges are drawn as thick lines. Changing the slope of these long edges moves all assignment pins above the horizontal line in one direction and all assignment pins below the horizontal line in the reverse direction. In this way, all connectors encode the same truth assignment of the underlying variable. Note that we can introduce as many connectors as needed for any one variable.

The positive clause gadget consists of a fixed boundary, a movable *wiggle* and three *assignment pin endings*. See Fig. 8, where the fixed boundary is shaded in gray, the wiggle is highlighted in blue and the assignment pin endings are highlighted in yellow.

The fixed boundary is connected to the reference pin of the variable gadget that is rightmost among the connected variable gadgets. Because the variable gadgets are fixed to the common frame, the boundary of the positive clause gadget is also connected to the common frame. The assignment pin endings are connected to assignment pins of connectors of the corresponding variable gadgets. The idea of the positive clause gadget is that the wiggle has to wiggle through the space bounded by the assignment pin endings on the left and the fixed boundary on the right. Recall that the assignment pins change their horizontal position depending on the truth assignment of their underlying variables. The positive clause gadget is designed so that the wiggle can always be drawn, except for the case when all variables are assigned to false. See Fig. 8a–c, which shows the three possible situations when exactly one variable is assigned to true. In any case where at least one variable is assigned to true the wiggle can be drawn in one of the three ways shown. However, as shown in Fig. 8d, the wiggle cannot be drawn in the case where all variables are assigned to false. The reason for this is that the the assignment pin endings get so close to the fixed boundary that they leave too little space for the wiggle to be drawn. This means that some vertices must intersect, for example those shown in red in Fig. 8d.

The negative clause gadgets works very similarly. It is drawn below the horizontal line of variables and it forces at least one of the incident variable gadgets to be configured as false. See Fig. 9 which shows admissible drawings (a–c) and that the case when all incident variables are configured as true cannot occur (d). Note that this uses the design of the variable gadget that the assignment and reference pins below the horizontal line are closer when the variable is assigned to true (i.e., the inverse situation compared to the situation above the horizontal line).

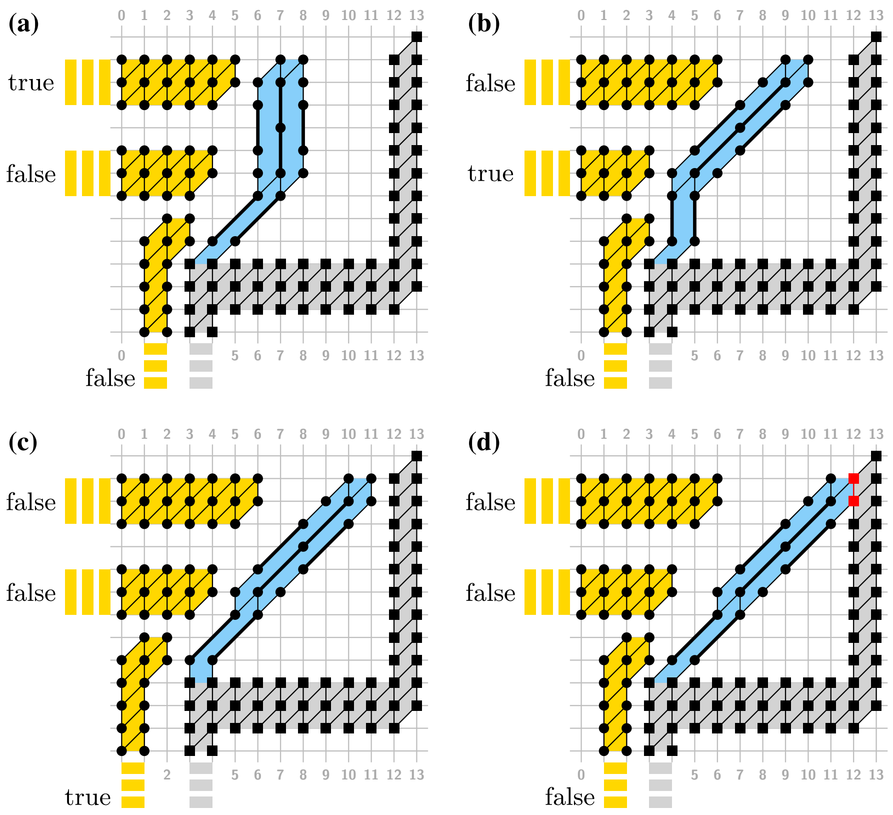

Fig. 8 The positive clause gadget when the drawing when at least one variable is assigned to true (a–c). No planar drawing exists when all variables are assigned to false (d), because this leads to intersections, e.g., at the vertices marked in red

It is evident that any level-planar two-slope drawing of the resulting graph induces a solution of the underlying planar monotone 3-SAT problem and vice versa. Note that the variable gadgets become biconnected when embedded into the common frame and that the clause gadgets are biconnected by design. Furthermore, all long edges have length two. We therefore conclude the following.

Theorem 5 λ-DRAWABILITY *is* NP*-complete even for* $\lambda = 2$ *and biconnected graphs where all edges have length one or two.*

7 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 compute 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 lets us find λ-

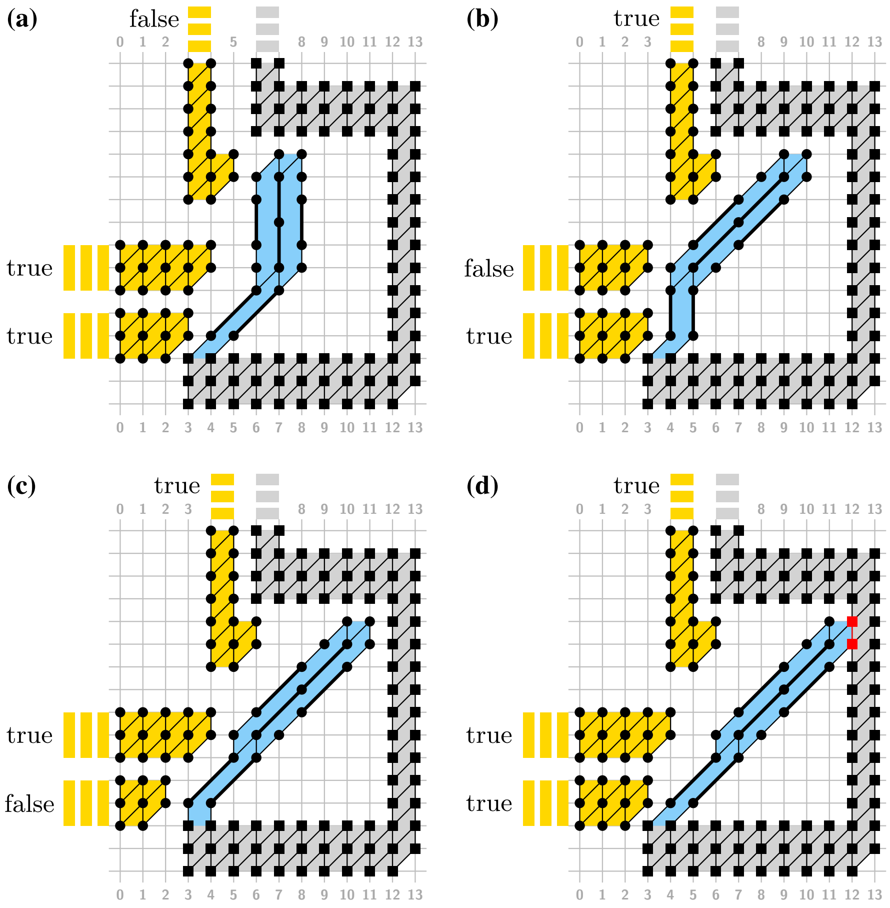

Fig. 9 The negative clause gadget when at least one variable is assigned to false (a–c). No drawing exists when all variables are assigned to true (d), because this leads to intersections, e.g., at the vertices marked in red. Note the symmetry to the positive clause gadget in Fig. 8

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. This leaves little room to extend our polynomial-time algorithms for more general classes of level-planar graphs with fixed embedding.

An interesting problem that remains open is that of finding λ-drawings when the embedding is not fixed. For orthogonal drawings, the SPQR-tree played a key role in going from optimizing a given fixed embedding [43] to optimizing across all possible embeddings of a graph [7]. A recent SPQR-tree-like embedding representation for level planarity [11] might enable a similar leap for λ-drawings.

Funding Open Access funding enabled and organized by Projekt DEAL.

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: Y. Hu, M. Nöllenburg (eds.) Graph Drawing and Network Visualization—24th International Symposium, GD 2016, Athens, Greece, September 19-21, 2016, Revised Selected Papers, *Lecture Notes in Computer Science*, vol. 9801, pp. 532–545. Springer (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: Cyclic, torus, and simultaneous level planarity. Theor. Comput. Sci. **804**, 161–170 (2020). https://doi.org/10.1016/j.tcs.2019.11.024
4. Barát, J., Matousek, J., Wood, D.R.: Bounded-degree graphs have arbitrarily large geometric thickness. Electron. J. Comb. **13**(1) (2006). https://doi.org/10.37236/1029
5. Bekos, M.A., Di Giacomo, E., Didimo, W., Liotta, G., Montecchiani, F.: Universal slope sets for upward planar drawings. In: T.C. Biedl, A. Kerren (eds.) Graph Drawing and Network Visualization - 26th International Symposium, GD 2018, Barcelona, Spain, September 26-28, 2018, Proceedings, *Lecture Notes in Computer Science*, vol. 11282, pp. 77–91. Springer (2018). https://doi.org/10.1007/978-3-030-04414-5_6
6. Bläsius, T., Kobourov, S.G., Rutter, I.: Simultaneous embedding of planar graphs. In: R. Tamassia (ed.) Handbook on Graph Drawing and Visualization, pp. 349–381. Chapman and Hall/CRC (2013)
7. Bläsius, T., Rutter, I., Wagner, D.: Optimal orthogonal graph drawing with convex bend costs. ACM Trans. Algorith. **12**(3), 33:1-33:32 (2016). https://doi.org/10.1145/2838736
8. 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. SIAM J. Comput. **46**(4), 1280–1303 (2017). https://doi.org/10.1137/15M1042929
9. Brückner, G., Krisam, N., Mchedlidze, T.: Level-planar drawings with few slopes. In: D. Archambault, C.D. Tóth (eds.) Graph Drawing and Network Visualization—27th International Symposium, GD 2019, Prague, Czech Republic, September 17-20, 2019, Proceedings, *Lecture Notes in Computer Science*, vol. 11904, pp. 559–572. Springer (2019). https://doi.org/10.1007/978-3-030-35802-0_42
10. Brückner, G., Rutter, I.: Partial and constrained level planarity. In: P.N. Klein (ed.) Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19, pp. 2000–2011. SIAM (2017). https://doi.org/10.1137/1.9781611974782.130
11. Brückner, G., Rutter, I.: An SPQR-tree-like embedding representation for level planarity. In: Y. Cao, S.W. Cheng, M. Li (eds.) 31st International Symposium on Algorithms and Computation, ISAAC 2020, December 14-18, 2020, Hong Kong, China (Virtual Conference), *LIPIcs*, vol. 181, pp. 8:1–8:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020). https://doi.org/10.4230/LIPIcs.ISAAC.2020.8
12. de Berg, M., Khosravi, A.: Optimal binary space partitions for segments in the plane. Int. J. Comput. Geom. Appl. **22**(3), 187–206 (2012). https://doi.org/10.1142/S0218195912500045. http://www.worldscinet.com/doi/abs/10.1142/S0218195912500045

13. Di Giacomo, E., Liotta, G., Montecchiani, F.: Drawing outer 1-planar graphs with few slopes. J. Graph Algorithms Appl. **19**(2), 707–741 (2015). https://doi.org/10.7155/jgaa.00376
14. 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). https://doi.org/10.1016/j.tcs.2017.12.004
15. Di Giacomo, E., Liotta, G., Montecchiani, F.: 1-bend upward planar slope number of SP-digraphs. Comput. Geom. **90**, 101628 (2020). https://doi.org/10.1016/j.comgeo.2020.101628
16. Dujmovic, V., Eppstein, D., Suderman, M., Wood, D.R.: Drawings of planar graphs with few slopes and segments. Comput. Geom. **38**(3), 194–212 (2007). https://doi.org/10.1016/j.comgeo.2006.09.002
17. Dujmovic, V., Suderman, M., Wood, D.R.: Graph drawings with few slopes. Comput. Geom. **38**(3), 181–193 (2007). https://doi.org/10.1016/j.comgeo.2006.08.002
18. Even, S., Itai, A., Shamir, A.: On the complexity of timetable and multicommodity flow problems. SIAM J. Comput. **5**(4), 691–703 (1976). https://doi.org/10.1137/0205048
19. Fakcharoenphol, J., Rao, S.: Planar graphs, negative weight edges, shortest paths, and near linear time. J. Comput. Syst. Sci. **72**(5), 868–889 (2006). https://doi.org/10.1016/j.jcss.2005.05.007
20. Hassin, R.: Maximum flow in (s, t) planar networks. Inf. Process. Lett. **13**(3), 107 (1981). https://doi.org/10.1016/0020-0190(81)90120-4
21. Healy, P., Nikolov, N.S.: Hierarchical drawing algorithms. In: R. Tamassia (ed.) Handbook on Graph Drawing and Visualization, pp. 409–453. Chapman and Hall/CRC (2013)
22. Henzinger, M., Klein, P.N., Rao, S., Subramanian, S.: Faster shortest-path algorithms for planar graphs. J. Comput. Syst. Sci. **55**(1), 3–23 (1997). https://doi.org/10.1006/jcss.1997.1493
23. Hoffmann, U.: On the complexity of the planar slope number problem. J. Graph Algorithms Appl. **21**(2), 183–193 (2017). https://doi.org/10.7155/jgaa.00411
24. Hu, T.C.: Integer Programming and Network Flows. Addison-Wesley, Reading, MA (1969)
25. Itai, A., Shiloach, Y.: Maximum flow in planar networks. SIAM J. Comput. **8**(2), 135–150 (1979). https://doi.org/10.1137/0208012
26. Jänicke, S., Geßner, A., Franzini, G., Terras, M., Mahony, S., Scheuermann, G.: Traviz: A visualization for variant graphs. Digit. Scholarsh. Humanit. **30**(Suppl-1), i83–i99 (2015). https://doi.org/10.1093/llc/fqv049
27. Keszegh, B., Pach, J., Pálvölgyi, D.: Drawing planar graphs of bounded degree with few slopes. SIAM J. Discret. Math. **27**(2), 1171–1183 (2013). https://doi.org/10.1137/100815001
28. Keszegh, B., Pach, J., Pálvölgyi, D., Tóth, G.: Drawing cubic graphs with at most five slopes. Comput. Geom. **40**(2), 138–147 (2008). https://doi.org/10.1016/j.comgeo.2007.05.003
29. 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). https://doi.org/10.7155/jgaa.00474
30. Kleinberg, J.M., Tardos, É.: Algorithm design. Addison-Wesley (2006)
31. Knauer, K.B., Micek, P., Walczak, B.: Outerplanar graph drawings with few slopes. Comput. Geom. **47**(5), 614–624 (2014). https://doi.org/10.1016/j.comgeo.2014.01.003
32. Lenhart, W., Liotta, G., Mondal, D., Nishat, R.I.: Planar and plane slope number of partial 2-trees. In: S.K. Wismath, A. Wolff (eds.) Graph Drawing - 21st International Symposium, GD 2013, Bordeaux, France, September 23-25, 2013, Revised Selected Papers, *Lecture Notes in Computer Science*, vol. 8242, pp. 412–423. Springer (2013). https://doi.org/10.1007/978-3-319-03841-4_36
33. Mchedlidze, T., Nöllenburg, M., Rutter, I.: Extending convex partial drawings of graphs. Algorithmica **76**(1), 47–67 (2016). https://doi.org/10.1007/s00453-015-0018-6
34. Meyers, C.A., Schulz, A.S.: Integer equal flows. Oper. Res. Lett. **37**(4), 245–249 (2009). https://doi.org/10.1016/j.orl.2009.03.006
35. Mozes, S., Wulff-Nilsen, C.: Shortest paths in planar graphs with real lengths in $o(n \log^2 n / \log\log n)$ time. In: M. de Berg, U. Meyer (eds.) Algorithms - ESA 2010, 18th Annual European Symposium, Liverpool, UK, September 6-8, 2010. Proceedings, Part II, *Lecture Notes in Computer Science*, vol. 6347, pp. 206–217. Springer (2010). https://doi.org/10.1007/978-3-642-15781-3_18
36. Nickel, S., Nöllenburg, M.: Towards data-driven multilinear metro maps. In: A. Pietarinen, P. Chapman, L. Bosveld-de Smet, V. Giardino, J.E. Corter, S. Linker (eds.) Diagrammatic Representation and Inference - 11th International Conference, Diagrams 2020, Tallinn, Estonia, August 24-28, 2020, Proceedings, *Lecture Notes in Computer Science*, vol. 12169, pp. 153–161. Springer (2020). https://doi.org/10.1007/978-3-030-54249-8_12

37. Nöllenburg, M.: A survey on automated metro map layout methods. In: Schematic Mapping Workshop. Essex, UK (2014)
38. Pach, J., Pálvölgyi, D.: Bounded-degree graphs can have arbitrarily large slope numbers. Electron. J. Comb. **13**(1) (2006). http://www.combinatorics.org/Volume_13/Abstracts/v13i1n1.html
39. Patrignani, M.: On extending a partial straight-line drawing. Int. J. Found. Comput. Sci. **17**(5), 1061–1070 (2006). https://doi.org/10.1142/S0129054106004261
40. Purchase, H.C.: Metrics for graph drawing aesthetics. J. Vis. Lang. Comput. **13**(5), 501–516 (2002). https://doi.org/10.1006/jvlc.2002.0232
41. Sahni, S.: Computationally related problems. SIAM J. Comput. **3**(4), 262–279 (1974). https://doi.org/10.1137/0203021
42. Srinathan, K., Goundan, P.R., Kumar, M.V.N.A., Nandakumar, R., Rangan, C.P.: Theory of equal-flows in networks. In: O.H. Ibarra, L. Zhang (eds.) Computing and Combinatorics, 8th Annual International Conference, COCOON 2002, Singapore, August 15-17, 2002, Proceedings, *Lecture Notes in Computer Science*, vol. 2387, pp. 514–524. Springer (2002). https://doi.org/10.1007/3-540-45655-4_55

43. Tamassia, R.: On embedding a graph in the grid with the minimum number of bends. SIAM J. Comput. **16**(3), 421–444 (1987). https://doi.org/10.1137/0216030