Drawing of Level Planar Graphs with Fixed Slopes

Bachelor Thesis of

Nadine Davina Krisam

At the Department of Informatics
Institute of Theoretical Computer Science

Reviewers:	Prof. Dr. Dorothea Wagner
	Prof. Dr. rer. nat. Peter Sanders
Advisors:	Guido Brückner, M.Sc.
	Dr. Tamara Mchedlidze

Time Period: May 1st, 2018 – August 31th, 2018

Statement of Authorship

I hereby declare that this document has been composed by myself and describes my own work, unless otherwise acknowledged in the text. I also declare that I have read the *Satzung zur Sicherung guter wissenschaftlicher Praxis am Karlsruher Institut für Technologie (KIT)*.

Karlsruhe, August 30, 2018

Abstract

A drawing of a directed acyclic planar graph is called *upward planar* if all edges are represented by monotonically increasing non-intersecting curves. A directed acyclic planar graph is called *level planar* if it has an upward planar drawing such that all vertices that are assigned to the same level have the same y-coordinate. In this thesis we study level planar drawings with two fixed slopes of the edges and call them LP2-drawings. We present an algorithm that, given a level planar graph, decides whether it has an LP2-drawing with an additional requirement of rectangular inner faces. After that, we drop the requirement of rectangular inner faces and provide an algorithm that decides whether a general LP2-drawing exists. Both algorithms have polynomial running time. In the conclusion of the thesis, we give an outlook how to extend the latter algorithm to work on multiple fixed slopes.

Deutsche Zusammenfassung

Eine Zeichnung eines gerichten azyklischen planaren Graphen heißt *aufwärts-planar*, wenn alle Kanten durch monoton steigende, sich nicht schneidende, Kurven dargestellt werden. So ein gerichteter azyklischer planarer Graph wird *level-planar* genannt, wenn er eine aufwärts-planare Zeichnung hat, in der alle Knoten, die dem selben Level zugewiesen wurden, die gleiche y-Koordinate haben. In dieser Arbeit werden wir uns mit level-planaren Zeichnungen mit zwei festgelegten Steigungen beschäftigen. Diese nennen wir LP2-Zeichnungen. Wir stellen einen Algorithmus vor, der gegeben einen level-planaren Graphen entscheidet, ob es eine LP2-Zeichnung gibt, in der wir uns auf rechteckige innere Facetten beschränken. Danach heben wir diese Einschränkung wieder auf und präsentieren einen Algorithmus, welcher entscheidet, ob eine LP2-Zeichnung existiert. Beide Algorithmen haben polynomielle Laufzeit. In der Zusammenfassung der Arbeit geben wir einen Ausblick, wie der zweite Algorithmus erweitert werden kann, sodass er mit mehreren festgelegten Steigungen arbeitet.

Contents

1 Introduction **1**
1.1 Related Work 2

2 Preliminaries **5**

3 LP2-Drawings with Rectangular Faces **7**
3.1 Rectangular LP2-Drawings 7
3.2 2-SAT for Rectangular LP2-Drawings 10

4 General LP2-Drawings **13**
4.1 LP2-Drawings of *st*-Graphs 13
4.2 Flow Network for LP2-Drawings 19

5 Conclusion **27**

Bibliography **29**

1. Introduction

Directed acyclic graphs appear often in practice, for instance in models like PERT-diagrams, business process models, text-variant graphs or phylogenetic networks. Text-variant graphs for example work on different translations of sentences. Here, vertices represent words or subsentences, and edges are between vertices that represent words that belong to the same translation and are consecutive in the sentence. In phylogenetic networks, vertices represent different species. Edges represent non-transitive ancestry relations between species. Sometimes the vertices can be grouped by some parameters. In text-variant graphs, the vertices that represent words with the same meaning can be grouped together. In phylogenetic networks, species of the same time period can be put into one group.

When representing the graphs from these domains, it is common to place the vertices of the same group on the same horizontal (or vertical) line, to emphasize there similarity. Those lines are also called levels. Levels representing different groups are usually represented by parallel lines. If the graph is planar and every vertex is assigned to a level, we call this graph *level planar graph*. Level planar graphs that have only edges between consecutive levels are called *proper*. For the drawing of level planar graphs the y-coordinate of the vertices are already fixed by the level that the vertex is assigned to.

We want to restrict the drawings of level planar graphs to those that only use straight line for the edges. This restriction is often used to make drawings simpler to read. Also, if the maximum degree of a graph is bounded, then the graph can be drawn with only few slopes. For instance, in phylogenetic graphs we normally have only small maximum degree, therefore we want to restrict the slopes used in the drawings of the level planar graphs. We now want to find algorithms that, given a proper level planar graph, decide whether a drawing with previously fixed slopes exists, and then also provide one. In this thesis we restrict our attention to the slopes -1 and 1.

In Chapter 2, we give necessary definitions that are used afterwards. We also state some simple observations. In Chapter 3, we give an algorithm to decide whether there is a level planar drawing with two slopes and rectangular inner faces of an *st*-graph. The algorithm constructs such a drawing if it exists. The algorithm is based on 2-SAT and works in polynomial time. In Chapter 4, we provide an algorithm that does not restrict the shape of the faces to rectangles. This algorithm is based on flow networks. In the conclusion we observe that the algorithm of Chapter 4 can be extended to an algorithm that finds drawings with a different number of fixed equidistant slopes. We also discuss how our approaches can be applied to the extension of partial level planar drawings and simultaneous level planar drawings with few slopes.

1.1 Related Work

In general, every planar graph can be drawn with only straight lines [F́48]. As the *slope number* of a graph we define the minimum number of slopes that are necessary to draw the graph in the plane. If the maximum degree of a graph is $\Delta \geq 5$, then graphs with slope number $n^{1-\frac{8+\epsilon}{\Delta+4}}$ can be found [DSW07]. Also, there exists no upper bound for the slope number [BMW06]. Graphs that have maximum degree 3 can be drawn with a maximum number of five slopes [KPPT08]. If the maximum degree of a planar graph is bounded, then the graph can be drawn with only few slopes [KPP10]. For planar graphs Dujmović et al. [DESW07] proved many bounds for the slope number of different types of graphs, as trees, outerplanar graphs and 2- or 3-connected graphs. For example, let n be the number of vertices of a graph G and Δ be the maximum degree. If G is a tree, the slope number is $\left\lceil \frac{\Delta}{2} \right\rceil$. For a planar 3-connected graph G, they proved the slope number to be greater than n and less than $2n$.

In an upward planar drawing, directed acyclic planar graphs are drawn such that all edges are represented by monotonically increasing curves. The problem of testing whether a graph has such an upward planar drawing is considered in [GT95]. Finding those drawings is topic of papers like [BT88] and [FRA08]. In Hasse diagrams, upward drawings of directed acyclic graphs that are transitively reduced are used to draw a partial order. Drawings of orders are considered in [CPR90] and [Riv93].

The height of a vertex in upward drawings is usually interpreted as the importance of that vertex. Sometimes vertices of the same importance are combined to layers or levels of the graphs. Then, we speak of *level graphs*, as they were already introduced in the previous section. Drawings of level graphs such that the levels are represented as parallel horizontal lines and the hierarchical structure is preserved is topic of papers like [ELT96]. In [ELT96] the Degree Weighted Barycentre (DWB) algorithm was presented, which focuses on drawing the graph in such a way that not only the hierarchy is preserved, but that the drawing is also planar, convex and symmetric if possible.

In [STT81] and [HN13] the Sugiyama Framework is presented. This framework consists of multiple algorithms to construct a layered graph drawing out of a directed graph. It takes several steps to construct this drawing. First possible cycles are eliminated such that the smallest number of edges has to be changed. Then the vertices are placed on levels with some criteria as to find the least number of necessary levels. Next the ordering of the vertices on one level is chosen. This ordering should result in a small number of edges crossing. At last the vertices are assigned to coordinates on there level. Healy and Kuusik published algorithms for choosing the order of the vertices on one level in [HK04].

Comparing the Sugiyama Framework to the work in this thesis, the graphs that we consider are already acyclic. Also the assignment of the vertices to layers and the ordering of the vertices on one level is given as an input.

When fixing slopes in general graphs, often orthogonal drawings are considered. Those drawings restrict the slopes to horizontal and vertical lines, but also allow bends of the edges. If the edges must have zero bends those drawings are also called rectlinear drawings. In this thesis we mostly consider drawings with only slopes -1 and 1. The resulting drawings can be compared to rectlinear drawings. In [TTV91] and [PT95] some bounds and algorithms for orthogonal drawings were presented. Tamassia et al. [Tam87] described such an algorithm. Given a planar embedding of a graph G with maximum degree ≤ 4 the algorithm finds a planar representation of G on a grid with a minimum number of bends. This representation then can be transformed into a grid drawing. They did this by constructing a flow network, where they had a node for each vertex and each face of G. Two face-nodes have an arc connecting them, if the faces are adjacent. A vertex-node is connected to a face-node, if the vertex is incident to that face. The flow over the arcs between two adjacent face-nodes then corresponds to the number of bends at the edge separating the faces. The flow over the arcs between a vertex-node and a face-node corresponds to the angle between the two edges, that are both incident to the face and the vertex.

Some algorithms for drawing on a grid, like the one described above, first give a representation of the graph, that can later be transformed into a drawing of the graph. In contrast to a drawing, in a representation of a graph only the slopes of the edges are considered. The length of the edges and by that the positions or even the distances between two nodes are not fixed. In a level planar graph, on the other hand, the positions of the vertices are restricted to the level that the vertex is assigned to. Even more, fixing the slopes results in a drawing of the graph up to horizontal translation. Therefore, the flow network technique employed by Tamassia [Tam87] can not be adapted in a straight-forward way to level planar graphs with slopes -1 and 1.

2. Preliminaries

A *level planar graph* (G, ℓ) is a planar graph $G = (V, E)$ together with a level assignment $\ell \colon V \to \mathbb{L}$, where $\mathbb{L} = \{l_1, \dots, l_n\}$ is the set of *levels* and n is the number of levels.

For all levels $l \in \mathbb{L}$ we define an ordering τ_l of the vertices on l from left to right. Let V_l be the set of vertices x with $\ell(x) = l$, that is $V_l = \{x \in V \mid \ell(x) = l\}$, then $\tau_l \colon V_l \to \{1, \dots, \mid V_l \mid\}$ is a bijection. For a level planar graph (G, ℓ) this ordering $\tau = \{\tau_i \mid i \in \mathbb{L}\}$ gives us a *level planar embedding* of this graph. A level planar graph (G, ℓ) with a fixed embedding is called a *level plane graph*.

Considering τ, we map vertices to coordinates with $\Gamma_V \colon V \to \mathbb{R} \times \mathbb{Z}, v \mapsto (x_v, y_v)$ such that $y_v = \ell(v)$ for all $v \in V$ and such that $x_u < x_v$ for all vertices $u, v \in V$ with $\ell(u) = \ell(v)$ and $u <_\tau v$. Each edge uv is mapped by Γ_E to the line segment between $\Gamma_V(u)$ and $\Gamma_V(v)$. A *level drawing* of (G, ℓ) consists of an ordering τ and the mapping $\Gamma = (\Gamma_V, \Gamma_E)$.

An internal crossing of two different edges is an intersection of the corresponding line segments without the start and end points. A level drawing is *planar*, if the edges are drawn without internal crossings and distinct vertices have distinct positions. Such a drawing is called *level planar drawing*. We only consider *proper* graphs in this thesis, thus graphs that have edges only between vertices on adjacent levels.

A *level planar 2-slope drawing* (*LP2-drawing* for short) is a level planar drawing, where every edge has either slope 1 or slope -1.

In contrast to orthogonal representations of planar graphs with maximum degree ≤ 4 we do not distinguish between an LP2-drawing and an LP2-representation (see Section 1.1). The distances between vertices are already fixed by the fixed distance between levels, the fixed level assignment and the slope assignment for each edge.

Let (G, ℓ) be a level planar graph with an LP2-drawing and $N(x) := \{y \in V(G) \mid xy \in E(G)\}$ the neighborhood of a vertex x. The following observation can be made:

Observation 2.1. *Every vertex $x \in V(G)$ has at most four neighbors. Also the following inequalities hold for every vertex $x \in V(G)$:*

$$\#\{v \in V(G) \mid v \in N(x) \wedge L(v) = L(x) - 1\} \leq 2$$
$$\#\{v \in V(G) \mid v \in N(x) \wedge L(v) = L(x) + 1\} \leq 2,$$

We will only consider graphs that fulfill those necessary conditions for having an LP2-drawing.

In a level plane graph (G, ℓ) each edge is incident to two not necessarily distinct faces f_1 and f_2. We always orientate an edge from the lower to the upper level. With that, we define the *left face* (*right face*, respectively) of an edge e as the face, that lies to the left (right, respectively) of e when traversing e in the direction of its orientation.

For each inner face f the *boundary of f* is a set of edges, that have f either as there left or as there right face, but not as both, and a set of vertices, that are incident to those edges. We denote this boundary by B_f.

A *source* is a vertex with only outgoing edges and a *sink* a vertex with only incoming edges. An *st-graph* is a directed acyclic graph with a single source and a single sink. The orientation of a level planar graph is acyclic, where each edge is directed from the lower to the upper level. If a level planar graph with this orientation has a single source and a single sink, we call this graph *level planar st-graph.*

Let (G, ℓ) be a level planar *st*-graph. Note that G has a unique vertex s with $\ell(s) < \ell(v)$, for all $v \in V(G), v \neq s$ and a unique vertex t with $\ell(t) > \ell(v)$, for all $v \in V(G), v \neq t$. Also for every v there is a path P_v from s over v to t, where all edges of P are directed from a lower to a higher level.

Further each face f of G has exactly one vertex a, such that the level $\ell(a) < \ell(v), \forall v \in f, v \neq a$, otherwise the graph is not a level planar *st*-graph by definition. Denote this vertex as $a =: s_f$, the source of f. Also there is exactly one vertex b, such that the level $\ell(b) > \ell(v)$, for all $v \in f, v \neq b$. Denote it as $b =: t_f$, the sink of f.

In an *st*-graph we call the boundaries of a face *st-boundaries.* Those *st*-boundaries can be described further. For a face f all edges of f are within B_f and the boundary consists of two independent paths going from s_f to t_f. Two paths are independent, if they share only the first and the last vertex. We call the vertices that are on only one path *internal vertices.* Denote one path as $P^l(f)$ and the other as $P^r(f)$, so $B_f = P^l(f) \cup P^r(f)$. For each path there is exactly one vertex on each level between $\ell(s_f)$ and $\ell(t_f)$, because G is proper. By planarity of the graph, every internal vertex of path $P^l(f)$ lies to the left of the vertex of $P^r(f)$ that is on the same level.

The *length* of a path P is defined by the number of edges in it. We denote it with $|P|$. For a face f we define the length of f by $\text{len}(f) = |P^l(f)| = |P^r(f)|$. Let $P^l(f) = [e_1, e_2, \ldots, e_{\text{len}(f)}]$ and $P^r(f) = [g_1, g_2, \ldots, g_{\text{len}(f)}]$ with e_1 and g_1 the lowest edges on $P^l(f)$ and $P^r(f)$. The edges e_i and $g_{\text{len}(f)-i+1}$ are called *twin edges* for $i \in \{1, \ldots, \text{len}(f)\}$.

3. LP2-Drawings with Rectangular Faces

Let (G, ℓ) be a level planar graph with $G = (V, E)$ and a given fixed embedding. We want to provide an LP2-drawing of the graph or declare that there is none. Note that for an LP2-drawing the level distant is equidistant and that the only possible slopes for an edge are 1 and -1. Also recall that we only consider proper level planar graphs, so graphs where the edges are only between adjacent levels. A restriction of the stated problem is the question whether there is a *rectangular LP2-drawing*. That is an LP2-drawing where each inner faces is a rectangle. For the outer face we make no restriction.

In the following section we show some conditions that are satisfied for rectangular LP2-drawings. After that we present an algorithm based on 2-SAT that provides an answer to the question if we can find a rectangular LP2-drawing for a level planar *st*-graph.

3.1 Rectangular LP2-Drawings

In this section we study the question whether a given level planar *st*-graph has a rectangular LP2-drawing.

In a rectangular LP2-drawing the following two conditions are satisfied:

1. *Fork:* For every vertex u with two neighbors v and w on the same level, the slopes of uv and uw are fixed. Thus, for instance if v lies to the left of w and both v and w are on the level above u, then uv has slope -1 and uw has slope 1.
2. *Twin edge:* Twin edges have the same slope.

Figure 3.1 illustrates a drawing of a graph, where the way it has to be drawn is determined by the fork and the twin edge condition. Orange edges are edges where the slope is fixed by the fork condition and green edges those that are directly fixed by the twin edge condition. The dotted line connects two twin edges, that are fixed by the fork condition, but have different slopes and contradict thus the twin edge condition. There is no possibility to draw the graph of Figure 3.1 as a rectangular LP2-drawing.

An example of a second graph, that has an rectangular LP2-drawing and fulfills both conditions is given in Figure 3.2. Again orange edges are edges, that are fixed by the fork condition and green edges are fixed by the twin edge condition.

In the preliminaries we have already mentioned, that in an *st*-graph each inner face f is bounded by two paths $P^l(f)$ and $P^r(f)$. For rectangular faces we can further restrict these paths.

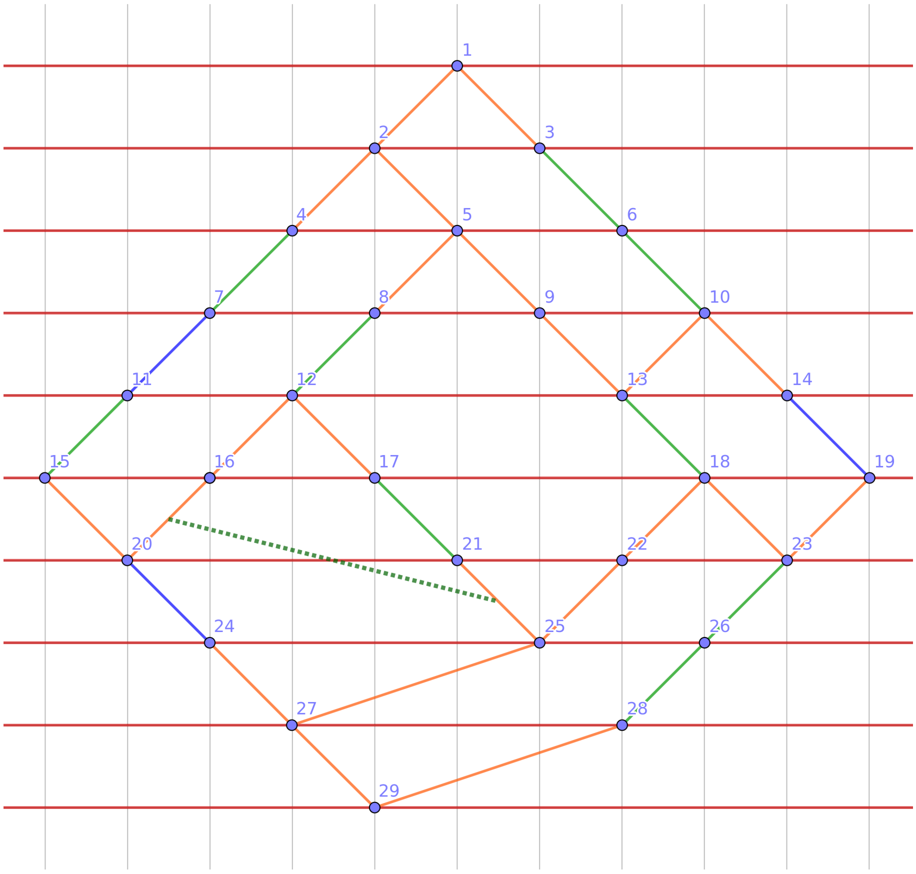


Figure 3.1: Example of a graph, that has no rectangular LP2-drawing

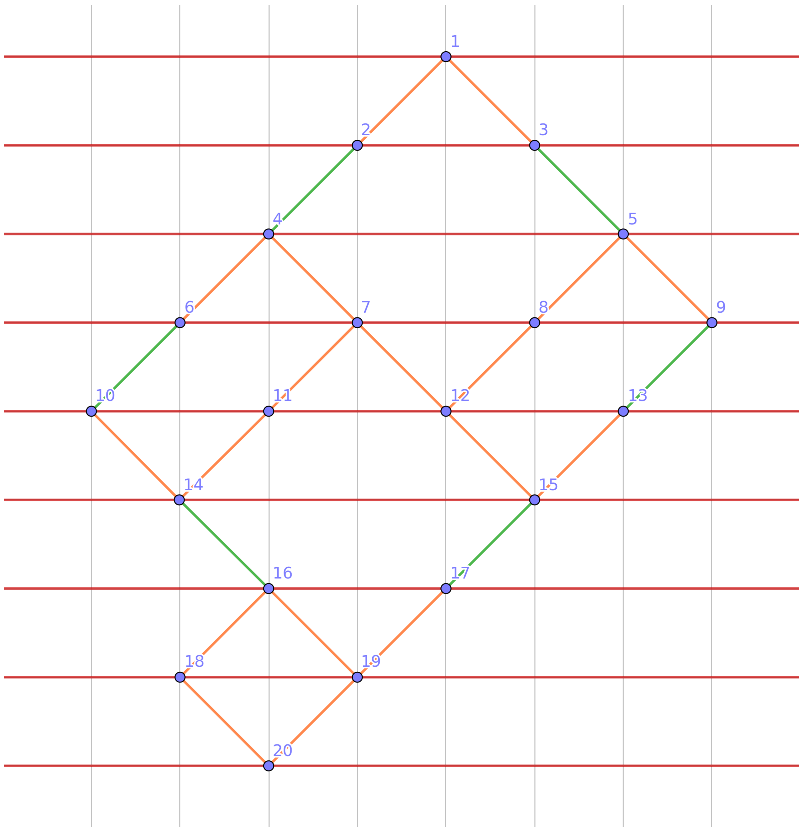


Figure 3.2: Example of an rectangular LP2-drawing of a graph

Lemma 3.1. *In an LP2-drawing of an st-graph an inner face is rectangular iff the left path consists of some edges with slope* -1 *followed by some edges with slope* 1 *and their twin edges have the same slope. Thus each path has exactly one bend and a left path has as many edges with slope* 1 *as the right path.*

Proof. A rectangle consists of four bends of 90 degrees. Two of those bends are fixed at the highest and lowest vertex of a face f. Those bends are diagonally opposite in the rectangle. Therefore there is exactly one change of slopes at each path of f. The fork condition already implies, that the first edge of a left path has slope -1 and the last edge has slope 1. Therefore on a left path, we first have some edges with slope -1 and the some edges with slope 1. The symmetry of a rectangle implies that twin edges have the same slope.

For the other direction of the equivalence we look at a face, with the left path consisting of some edges with slope -1 first and then of some edges with slope 1, and where twin edges have the same slope. We want to show, that this face is rectangular. We can divide the boundary of this face into four straight lines: two lines at the left path, the first having slope -1 and the second slope 1, and two lines at the right path, the first with slope 1 and the second with slope -1. Those lines form a quadrangle. Also all four bends have 90 degrees because we always change from slope 1 to -1 or from slope -1 to 1. Therefore the face is a rectangle. □

3.2 2-SAT for Rectangular LP2-Drawings

In this section we present an algorithm based on 2-SAT formulations that decides whether there exists a rectangular LP2-drawing and constructs one if it exists. Let (G, ℓ) be a level plane *st*-graph. So we already have given the level assignment and an embedding τ that fixes the ordering of the vertices on one level. The only thing that needs to be considered for a drawing is therefore the x-coordinate of every vertex. Instead of choosing the x-coordinates individually for every vertex, we will choose the slopes of every edge. Then a drawing is fixed up to translation along the x-axis. For choosing the slopes we use the fork and the twin edge condition as defined in the previous section. They restrict possible slopes for edges. Also some slopes influence each other. We will represent this dependencies through clauses. The resulting Boolean formula will be proved to be satisfiable if and only if there exists an rectangular LP2-drawing.

For each inner face f of G and each edge $e \in f$ we add a Boolean *variable* $s(f, e)$. Intuitively this variables can be identified with the slope that e has with respect to a face f by the following rule:

$$s(f, e) = 0 \Leftrightarrow \text{slope}(e) = \text{-}1$$
$$s(f, e) = 1 \Leftrightarrow \text{slope}(e) = 1$$

Thus each edge is represented by two variables corresponding to the two faces the edge bounds.

Next for all inner faces f of G there is a list of *clauses* that will be added to the 2-SAT instance. These clauses ensure that each face is rectangular and because of that, opposite sides of this rectangle have the same length. Let $P^l(f) = [e_1, e_2, \ldots, e_{\text{len}(f)}]$ be the edges of the left path of f. We now construct a set of clauses as follows:

Cl.1 $s(f, e_i) \rightarrow s(f, e_{i+1})$ with $i \in \{1, \ldots, \text{len}(f) - 1\}$[1]

Cl.2 $\forall e \in P^l(f)$ and the twin edge g of e in terms of f: $s(f, e) \leftrightarrow s(f, g)$[2]

Cl.3 $s(f, e_1) \leftrightarrow 0$

Cl.4 $s(f, e_{len(f)}) \leftrightarrow 1$

Because of the twin edges the clauses for the left path restrict the slopes for the right path, too.

Lemma 3.2. *Let f be an inner face of G and φ an assignment of the variables. If Clauses Cl.1 to Cl.4 are satisfied for f then φ corresponds to a rectangular drawing of f.*

Proof. The Clause Cl.3 and Cl.4 ensure, that there is at least one bend on each path. Without loss of generality we only look at the left path. There are two consecutive edges e_i and e_{i+1} along the path, where $s(f, e_i) = 0$ and $s(f, e_{i+1}) = 1$. That means that e_i is below and e_{i+1} above a bend and thus, there is a bend at the vertex v, that is incident to both e_i and e_{i+1}.

Also there cannot be two bends at one path. Suppose there is a path with two bends. Then there is an edge e_j, directly above the lowest bend with $s(f, e_j) = 1$, and an edge e_k directly below the next bend. Edge e_k needs to have $s(f, e_k) = 0$, otherwise there was no bend. But then $j < k$ and Clause Cl.1 $s(f, e_i) \rightarrow s(f, e_{i+1})$ cannot be satisfied for some $i \in \{l, \ldots, k - 1\}$, which is a contradiction. Thus, each path has exactly one bend, and that means the face is a quadrangle.

[1] An implication ($\rightarrow$) can be transformed in a 2-literal clause: $a \rightarrow b = \neg a \vee b$.
[2] An equivalency ($\leftrightarrow$) can be transformed into $a \leftrightarrow b = (a \rightarrow b) \wedge (b \rightarrow a)$.

Opposite sides of a face have the same length because twin edges have the same slope by Clause Cl.2 and the distances between levels are the same. Therefore by assigning each edge e the slope that is implied by $s(f,e)$ and fixing the coordinates of one vertex arbitrarily, the drawing of the inner face f is a rectangle. □

Let f and h be two inner faces that share an edge e. The following clause makes sure that the assignment of the slope of e at f and h are consistent. For every two inner faces f, h and every edge e with $e \in f, h$ we add the clause:

Cl.5 $s(f,e) \leftrightarrow s(h,e)$

That completes the Boolean formula to describe a rectangular LP2-drawing, as proven in the following theorem. Given a level plane st-graph (G,ℓ), let $\mathcal{S}(G,\ell)$ be the 2-SAT instance that is formed by Clause Cl.1 to Clause Cl.4 for all faces of G and Clause Cl.5 for all edges of G.

Theorem 3.3. *Let (G,ℓ) be a level plane st-graph and $\mathcal{S} = \mathcal{S}(G,\ell)$. Every assignment of the variables, that satisfies $\mathcal{S}$ corresponds bijectively to a rectangular LP2-drawing.*

Proof. Let (G,ℓ) be a level plane st-graph and let $\mathcal{S} = \mathcal{S}(G,\ell)$ be the 2-SAT instance.

" $\Leftarrow$ " We show, that if there is a rectangular LP2-drawing Γ, then there is an assignment, that satisfies $\mathcal{S}$. For each edge $e \in G$ and each face f that e is incident to, we set

$$s(e,f) = \begin{cases} 0 & \text{if slope}(e) = \text{-}1 \\ 1 & \text{if slope}(e) = 1 \end{cases}$$

and get an assignment φ of $\mathcal{S}$. We can show that φ satisfies $\mathcal{S}$, by proving that φ satisfies each clause.

Let f be some inner face, $P^l(f)$ the left path from s_f to t_f and $P^r(f)$ the right path. Vertex s_f has two neighbors on the level above, say u and w. Let u be to the left of w by τ without loss of generality. Because Γ is an LP2-drawing, the edge $s_f u$ has slope -1 and $s_f w$ slope 1, otherwise w would be drawn to the left of v. Therefore Clause Cl.3 is satisfied. The same proof can be made for Clause Cl.4.

Clause Cl.1 is fulfilled, because of Lemma 3.1 there is exactly one bend at each path and edges on $P^l(f)$ first go to the left and after the bend to the right. With that Clause Cl.1 is satisfied for each edge on a left path. Clause Cl.2 is satisfied, because in a rectangle the two opposite sides have the same length. Because every edge has the same fixed length, twin edges have the same slope. That every edge has only one slope, regardless the face we look at, is given by Clause Cl.5.

" $\Rightarrow$ " It must be shown that an assignment φ of the variables that satisfies $\mathcal{S}$ can be converted into a rectangular LP2-drawing of G. So let φ be a satisfiable assignment. We give every edge e the slope that the assignment implies and fix the x-coordinates of s, the lowest vertex of G, arbitrarily. The y-coordinate is already fixed by the level that s is on. Thus the slopes of the edges of G give us a complete drawing. Now show that the resulting drawing Γ is a rectangular LP2-drawing of G.

By Lemma 3.2 each inner face is a rectangle. Because (G,ℓ) is an st-graph, there is a vertex s with $\ell(s) < \ell(v)$, for every $v \in V(G), v \neq s$. We will now construct the drawing level by level, starting at $\ell(s)$, and ensure the following invariants, where l denotes the considered level.

I.1 From left to right l first cuts the outer face, then some inner faces and at last the outer face again.

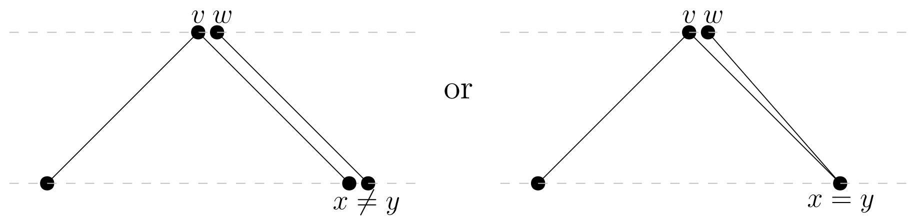


Figure 3.3: Different possibilities for the neighbors of v and w in Case 1 of Theorem 3.3

I.2 Edges have no internal crossings.

I.3 Distinct vertices on the same level have distinct x-coordinates.

Let all invariants be true for every level below the currently considered level l. We now need to prove that the invariants hold for l, too. The first invariant I.3 is true for every level because we only look at *st*-graphs. If the outer face was between two inner faces, then there would be at least two sources or two sinks. The last two invariants I.2 and I.3 will ensure, that the drawing of the graph up to l is planar.

In an *st*-graph the distance between two vertices v and w on a level is even. That is true because we can find an s-v-path and an s-w-path. Each vertex on such a path cannot lie at the same x-coordinate as the vertex before it but lies one unit to the left or to the right. Because both paths start at the same point s, the distance between v and w is even. If edges have an internal crossing, then the distance between the vertices on both levels is 1, because there are only edges between two adjacent levels. Thus Invariant I.2 is true.

To prove Invariant I.3, we make a proof by contradiction. Consider two different vertices v and w occupying the same space at l. Each of the vertices has at most one neighbor on the level below, because of the *st*-graph. Next we make a case distinction whether v and w have two or one lower neighbors:

In Case 1 at least one vertex has two lower neighbors, without loss of generality v. One of this neighbors x of v occupies the same space as the lower neighbor of w, call it y. In Figure 3.3 the different possibilities for x and y are shown. If $x \neq y$, there is a contradiction of the third invariant at the level before. If $x = y$, then both v and w are upper neighbors of x, and thus they form the lower part of an inner face. Then the clauses ensure that v and w occupy different places at l, because this face is a rectangle. Therefore this case can never happen.

In Case 2 both v and w have exactly one lower neighbor, say v has x and w has y as lower neighbor. We know that the distance between x and y is 2, and that there in no other vertex between those two. Therefore x and y and thus v and w share a face f. Face f is an inner face, but it can not be a rectangle, if $v \neq w$ because either both v and w are t_f or there is more then one bend on both paths from s_f to t_f. But then φ is not a proper assignment of $\mathcal{S}$, because Lemma 3.2 already ensures, that each inner face is a rectangle. Therefore there are no two different vertices occupying the same space at a level. Thus every invariant holds for every level and results in a rectangular LP2-drawing of the graph.

□

Now we have proven that determining whether a level plane graph has a rectangular LP2-drawing can be encoded as a 2-SAT instance. An assignment of a 2-SAT instance can be found in polynomial time, so with the here presented method a rectangular LP2-drawing can also be found in polynomial time.

4. General LP2-Drawings

In this chapter we describe algorithms that given a level plane graph decide if it has an LP2-drawing. For those algorithms we use flow networks. In comparison to the previous chapter we drop the restriction of rectangular inner faces.

4.1 LP2-Drawings of st-Graphs

We first consider only st-graphs and discuss non-st-graph in the next section. So let (G, ℓ) be a level plane st-graph. With that restriction the boundary of every face is an st-boundary, too. Since the considered graphs are always proper, there are exactly two vertices of one face at every level between the highest and the lowest level of a face.

For each inner face f and a drawing Γ_f we define the *width of f at a level l_i* as the distance between the two vertices of level l_i. We define $\text{width}_i(f) = x_w - x_v$, where x_v, x_w are the x-coordinates of $v \in P^l(f)$ and $w \in P^r(f)$ and $\ell(v) = \ell(w) = l_i$. For the highest and lowest level of f we define the width to be 0. The distance between two vertices at the same level is even, because we only allow slopes -1 and 1.

Given a drawing of an inner face f with slopes -1 and 1, we denote by $n_i^l(f)$ and $n_i^r(f)$ the *number of edges with slope* 1 *in* f that are below the level l_i and lie on $P^l(f)$ and $P^r(f)$, respectively.

Lemma 4.1. *Let G be a level plane st-graph, f an inner face of G and Γ_f an LP2-drawing of f. For every level l_i with vertices of f on l_i the following equation holds:*

$$\text{width}_i(f) = 2 \cdot (n_i^r(f) - n_i^l(f)).$$

Proof. Let Γ_f be a drawing of f. If $l_i = \ell(s_f)$, then we have $\text{width}_i(f) = 0$ by definition. We also have $n_i^l(f) = n_i^r(f) = 0$, because we have no edges below level l_i, that are incident to f.

Now consider level l_i with $\ell(s_f) < l_i < \ell(t_f)$. Let v and w be the two vertices on l_i that are incident to f. Without loss of generality we have $v \in P^l(f)$ and $w \in P^r(f)$. Let x_v be the x-coordinate of v and x_w the x-coordinate of w. Then $\text{width}_i(f) = x_w - x_v$.

Through the s_f-v-path $P_1 \subsetneq P^l(f)$ and the s_f-w-path $P_2 \subsetneq P^r(f)$ the coordinates x_v and x_w are fixed with respect to the x-coordinate x_s of s_f. We have

$$x_v = x_s + n_i^l(f) - (\ell(v) - \ell(s_f) - n_i^l(f)) = x_s + 2 \cdot n_i^l(f) + \ell(s_f) - i \text{ and}$$

$$x_w = x_s + n_i^r(f) + (\ell(w) - \ell(s_f) - n_i^r(f)) = x_s + 2 \cdot n_i^r(f) + \ell(s_f) - i,$$

because the distance between two adjacent levels is always 1 and we only allow slopes -1 and 1. Thus we have

$$\text{width}_i(f) = x_w - x_v = (x_s + 2 \cdot n_i^r(f) + \ell(s_f) - i) - (x_s + 2 \cdot n_i^l(f) + \ell(s_f) - i) = 2 \cdot (n_i^r(f) - n_i^l(f)).$$

For level $l_i = \ell(t_f)$ we have defined the width as $\text{width}_i(f) = 0$ We also need the x-coordinate x_t of t_f to be the same whether we look at $P^l(f)$ or $P^r(f)$. So $x_t = x_s + 2 \cdot n_i^l(f) + \ell(s_f) - \ell(t_f) = x_s + 2 \cdot n_i^r(f) + \ell(s_f) - \ell(t_f)$ and thus $n_i^l(f) = n_i^r(f)$. But then we have $2 \cdot (n_i^r(f) - n_i^l(f)) = 0$. □

Observe that since $|P^l(f)| = |P^r(f)|$, f has an equal number of edges with slope -1 on $P^l(f)$ and $P^r(f)$. For a drawing of a face to be planar, we need that we have no internal crossings of edges and no two distinct vertices to be assigned to the same coordinates. It is easy to see that this is equivalent to the width of the face to be strictly greater than zero for every level that is not the highest or the lowest. The following Lemma follows from Lemma 4.1.

Lemma 4.2. *Let G be a level plane st-graph and f an inner face of G. For a drawing Γ_f of f with slopes* -1 *and* 1 *the following statements are equivalent:*

(i) Γ_f *is an LP2-drawing,*

(ii) $\text{width}_i(f) > 0$ *for every level l_i with $\ell(s_f) < l_i < \ell(t_f)$,*

(iii) for level $l_i \in \{\ell(s_f), \ell(t_f)\}$ we have $n_i^l(f) = n_i^r(f)$ and for level $\ell(s_f) < l_j < \ell(t_f)$ we have $n_j^l(f) < n_j^r(f)$.

Next, we define a flow network, motivated by the definitions used by Tamassia in [Tam87]. A *network* $\mathcal{N}$ consists of a set N of *nodes*, and a set A of *arcs*. Every arc (u, v) has a *lower bound* $0 \leq \lambda(u, v)$ and a *capacity* (upper bound) $\mu(u, v)$. A *flow* Φ of $\mathcal{N} = (N, A)$ is a function $\Phi\colon A \to \mathbb{N}_0$ that assigns an integer $\Phi(u, v)$ to every arc (u, v). The flow Φ is valid if the following holds:

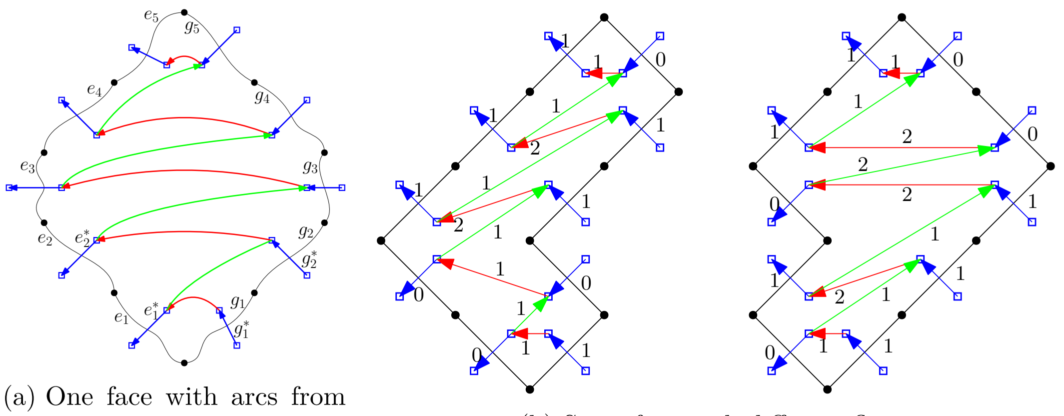
1. For every arc (u, v) the value $\Phi(u, v)$ satisfies $\lambda(u, v) \leq \Phi(u, v) \leq \mu(u, v)$.
2. For every node, that is not a source or a sink, the sum of the flows of incoming and outgoing arcs is equal.

Observe that then sum of the outgoing flow of all sources is equal to the sum of the incoming flow of all sinks.

Now we construct such a network $\mathcal{N}_G$ that corresponds to a level planar *st*-graph (G, ℓ) with a fixed embedding. For every edge e in G the network $\mathcal{N}_G$ has a dual arc e^* going from a left node $r[e^*]$ to a right node $l[e^*]$. In Figure 4.1 these are the blue edges. Intuitively, flow of value 1 going through those arcs corresponds to a slope of 1 for the corresponding primal edge and a flow of 0 to a slope of -1. Therefore we set the lower bound $\lambda(e^*) = 0$ and the capacity $\mu(e^*) = 1$ for a dual arc e^* of an edge e. For a drawing that is induced by a valid flow and an edge e, we set:

$$\text{slope}(e) = \begin{cases} \text{-1} & \text{if } \Phi(e^*) = 0 \\ 1 & \text{if } \Phi(e^*) = 1 \end{cases}$$

Next, for every inner face f we add arcs between the right nodes of edges on $P^l(f)$ and the left nodes of edges on $P^r(f)$. For this let $P^l(f) = [e_1, \ldots, e_{\text{len}(f)}]$ and $P^r(f) = [g_1, \ldots, g_{\text{len}(f)}]$.



(a) One face with arcs from left to right

(b) Same face with different flows

Figure 4.1: Network and flows for one face

We add arcs $(l[g_i^*], r[e_i^*])$ for every $i \in \{1, \ldots, \text{len}(f)\}$. These are the red arcs in Figure 4.1. Also we add arcs $(r[e_j^*], l[g_{j+1}^*])$, the green arcs, for every $j \in \{1, \ldots, \text{len}(f)-1\}$. For the sake of brevity we refer to those arcs by $(g_i, e_i) = (l[g_i^*], r[e_i^*])$ and by $(e_j, g_{j+1}) = (r[e_j^*], l[g_{j+1}^*])$. If we have an arc (u, v) of one of these types, then we set $\lambda(u, v) = 1$ and $\mu(u, v) = \infty$. The flow over those arcs corresponds to the width of a face and thus the lower bound 1 ensures, that the distance between two distinct vertices is nonzero. We call them *internal arcs* since they are always in exactly one face.

At last, we add one source s^* and arcs $(s^*, r[g^*])$ for every edge g that has the outer face as right face. Also we add one sink t^* and arcs $(l[e^*], t^*)$ for every edge e that has the outer face as left face. All those arcs (u, v) have $\lambda(u, v) = 0$ and $\mu(u, v) = 1$, as they are dual arcs and the flow over them corresponds to the slope of their primal edge, as described above. Observe that the constructed network has exactly one source and one sink.

Now we fix an arbitrarily face f of G. For f let $P^l(f) = [e_1, \ldots, e_{\text{len}(f)}]$ and $P^r(f) = [g_1, \ldots, g_{\text{len}(f)}]$. A valid flow Φ of $\mathcal{N}_G$ *induces* a drawing Γ_f of f as follows. Consider a fixed position of s_f, the lowest vertex of f. The slopes of the edges of $P^l(f)$ and $P^r(f)$ corresponding to the flow on their dual edges, determine the position of the vertices of f. The next lemmas prove an equivalence between the flow Φ and the induced drawing Γ_f.

For a fixed inner face f we denote with l_i for $i \in \{0, \ldots, \text{len}(f)\}$ the levels which contain vertices of f. With l_0 we denote the lowest level and with $l_{\text{len}(f)}$ the highest level of f. We say an arc (u, v) in the face f *crosses* a level l_i for $i \in \{1, \ldots, \text{len}(f)-1\}$, if $u = r[e_i]$ and $v = l[g_{i+1}]$.

Lemma 4.3. *Let G be a level plane st-graph, f an inner face of G, Φ is a valid flow of $\mathcal{N}_G$ and Γ_f an induced drawing of f. For every level l_i with $i \in \{1, \ldots, \text{len}(f)-1\}$ the following equation holds for an arc (u, v) that crosses l_i:*

$$\Phi(u, v) = n_i^r(f) - n_i^l(f).$$

Proof. By construction of Γ_f, an edge has slope 1 if and only if the flow going through the dual arc is 1. The flow conservation gives us that every unit of flow incoming to f through an arc g_j^* with $j \leq i$ either goes out through an arc e_k^* with $k \leq i$ or will be passed upwards through the arc $(u, v) = (e_i, g_{i+1})$, that crosses l_i. Thus the flow value of (u, v) is:

$$\Phi(u, v) = \sum_{j=1}^{i} \Phi(g_j^*) - \sum_{k=1}^{i} \Phi(e_k^*).$$

Since $\Phi(g_j^*)$ and $\Phi(e_k^*)$ take values in $\{0, 1\}$ instead of the sum of the flows we can count the number of arcs with with flow 1 beneath level l_i. Those arcs then correspond to edges with slope 1 that are beneath level l_i. Thus we have:

$$\Phi(u, v) = \#\{g_j \mid \Phi(g_j^*) = 1 \text{ and } j \leq i\} - \#\{e_k \mid \Phi(e_k^*) = 1 \text{ and } k \leq i\} \Leftrightarrow$$

$$\Phi(u, v) = \#\{g_j \mid \text{slope}(g_j) = 1 \text{ and } j \leq i\} - \#\{e_k \mid \text{slope}(e_k) = 1 \text{ and } k \leq i\} \Leftrightarrow$$

$$\Phi(u, v) = n_i^r(f) - n_i^l(f).$$

Therefore for every level l_i with $i \in \{1, \ldots, \text{len}(f) - 1\}$ and the arc (u, v) that crosses l_i the equation $\Phi(u, v) = n_i^r(f) - n_i^l(f)$ holds for every flow Φ and the induced by it drawing Γ_f of a face f. □

Now, we show that Γ_f is an LP2-drawing of f, if Γ_f is induced by a valid flow Φ. For this let the *incoming flow of f* (*outgoing flow of f*, respectively) be the sum of flow that goes through arcs g_i^* (e_i^*, respectively) for $i \in \{1, \ldots, \text{len}(f)\}$.

Lemma 4.4. *A valid flow Φ of $\mathcal{N}_G$ induces an LP2-drawing of f.*

Proof. Let Φ be a valid flow of $\mathcal{N}_G$ and Γ_f the induced drawing of f. We prove that Γ_f is an LP2-drawing. For this we prove that for level $l_i \in \{\ell(s_f), \ell(t_f)\}$ we have $n_i^l(f) = n_i^r(f)$ and for level $\ell(s_f) < l_j < \ell(t_f)$ we have $n_j^l(f) < n_j^r(f)$.

For $l_i = \ell(s_f)$ we have $n_i^l(f) = n_i^r(f) = 0$. So consider a level l_i with $\ell(s_f) < l_i < \ell(t_f)$. We know that the arc $(u, v) = (e_i, g_{i+1})$ that crosses l_i has lower bound $\lambda(u, v) = 1$ and that $\Phi(u, v) = n_i^r(f) - n_i^l(f)$, because of Lemma 4.3. Thus $1 \leq n_i^r(f) - n_i^l(f)$ and with that $n_i^l(f) < n_i^r(f)$. For $l_i = \ell(t_f)$ let $a = n_i^r(f)$ and $b = n_i^l(f)$ be the number of edges on $P^r(f)$ and $P^l(f)$ respectively with slope 1. The flow that goes through the dual arc of one of those edges is 1. Note that the whole incoming flow of f goes through the k arcs that are dual to the a edges on $P^r(f)$. So the value of the incoming flow of f is a. Also the whole outgoing flow of f goes through the b arcs that are dual to the b edges on $P^l(f)$ and thus the value of the outgoing flow of f is b. Because the flow conservation is satisfied, we have $a = b$.

Then by Lemma 4.2 Γ_f is an LP2-drawing. □

In the following, we aim to generalize the statement of Lemma 4.4 to the whole graph G. In analogy to a by Φ induced drawing of a single face f, we define an induced drawing of the whole graph (G, ℓ). Assume s, the lowest vertex of G, has a fixed coordinate. Again, let the slope of an edge e correspond to the flow Φ over edge e^*. Every vertex v in G is assigned to the coordinates, that are fixed by the coordinates of s and the slopes on a s-v-path. First, we show that Γ is well defined if Φ is valid, so that every vertex gets the same coordinates regardless what s-v-path we choose. For this we use that every edge $e \in G$ has the same slope slope(e), regardless whether we look from the left or the right face of e. Thus drawings of faces can be added to one another to create the induced drawing of (G, ℓ). Recall that the coordinates of the vertices are unique for each face. Thus the coordinates of one vertex are unique in Γ, too, and hence Γ is well-defined.

Next we show that a valid flow over $\mathcal{N}_G$ gives us an LP2-drawing of (G, ℓ) and that for every LP2-drawing of (G, ℓ) there is a valid flow over $\mathcal{N}_G$.

Theorem 4.5. *Let (G, ℓ) be a level plane st-graph and $\mathcal{N}_G$ the corresponding flow network. Every flow Φ of $\mathcal{N}_G$ corresponds bijectively to an LP2-drawing Γ of (G, ℓ).*

Proof. Let (G, ℓ) be a level plane *st*-graph and let $\mathcal{N}_G$ be the corresponding flow network.

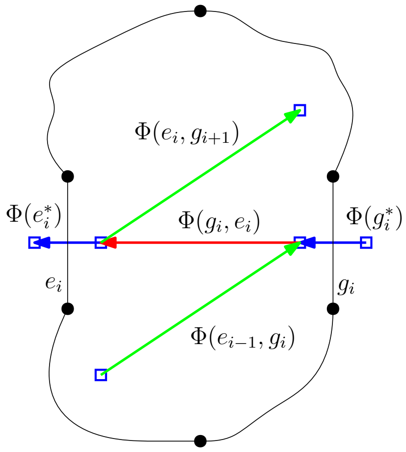


Figure 4.2: Illustration for assigning the correct flow to an arc (g_i, e_i)

"$\Leftarrow$ " We show, that if there is an LP2-drawing Γ of (G, ℓ), then there is a valid flow over $\mathcal{N}_G$. So we need to give each arc (u, v) a flow value $\Phi(u, v)$ such that the resulting flow Φ is a valid flow of $\mathcal{N}_G$. First set the flow at dual arcs as follows.

$$\Phi(e^*) = \begin{cases} 0 & \text{if slope}(e) = \text{-1} \\ 1 & \text{if slope}(e) = 1. \end{cases}$$

Next, we consider the internal arcs. For this we fix an arbitrary face f. By Lemma 4.2, we have $n_i^l(f) = n_i^r(f)$ for level $l_i = \ell(t_f)$ and every inner face f. Thus we have as many right nodes with incoming flow 1 corresponding to edges on $P^r(f)$ as we have left nodes with outgoing flow 1 corresponding to edges on $P^l(f)$. Recall that the arc between nodes $l[g_i^*]$ and $r[e_i^*]$ (or $r[e_j^*]$ and $l[g_{j+1}^*]$) is denoted (g_i, e_i) ((e_j, g_{j+1}), resp.), for $i \in \{1, \ldots, \text{len}(f)\}$ ($j \in \{1, \ldots, \text{len}(f) - 1\}$, resp.).

First we set $\Phi(g_1, e_1) = 1$ and $\Phi(e_1, g_2) = 1$. This is fixed for every drawing of every face, because slope$(g_1) = 1$ and slope$(e_1) =$ -1 and thus the flow conservation at $r[e_1^*]$ and $l[g_1^*]$ is satisfied.

Next we set $\Phi(e_j, g_{j+1}) = n_j^r(f) - n_j^l(f)$. Then we have $\Phi(e_j, g_{j+1}) \geq \lambda(e_j, g_{j+1}) = 1$ because $n_j^l(f) < n_j^r(f)$ by Lemma 4.2.

For every other arc (g_i, e_i) for $i \in \{2, \ldots, \text{len}(f) - 1\}$ we set the flow such that the flow conservation is fulfilled. The arc (g_i, e_i) is the only outgoing arc of a left node $l[g_i^*]$ and the only incoming arc of of a right node $r[e_i^*]$, as illustrated in Figure 4.2. We need to prove that

$$\Phi(e_{i-1}, g_i) + \Phi(g_i^*) = \Phi(e_i^*) + \Phi(e_i, g_{i+1}). \tag{4.1}$$

Then we can set $\Phi(g_i, e_i) = \Phi(e_{i-1}, g_i) + \Phi(g_i^*)$ and the flow conservation at $l[g_i^*]$ and $r[e_i^*]$ is fulfilled.

To prove equation (4.1) we make a case analysis on the slopes of e_i and g_i.

Case 1: slope$(e_i) =$ slope(g_i)**:** Then also $n_i^r(f) - n_i^l(f) = n_{i+1}^r(f) - n_{i+1}^l(f)$ and thus $\Phi(e_{i-1}, g_i) = \Phi(e_i, g_{i+1})$. Thus, the statement is true for this case.

Case 2: slope$(e_i) =$ **-1 and** slope$(g_i) = 1$**:** Then

$$\begin{aligned} n_i^r(f) - n_i^l(f) + 1 = n_{i+1}^r(f) - n_{i+1}^l(f) &\Leftrightarrow \\ \Phi(e_{i-1}, g_i) + 1 = \Phi(e_i, g_{i+1}) &\Leftrightarrow \\ \Phi(e_{i-1}, g_i) + \Phi(g_i^*) = \Phi(e_i, g_{i+1}) + \Phi(e_i^*). \end{aligned}$$

Case 3: slope(e_i) = 1 **and** slope(g_i) = **-1:** In this case we have

$$n_i^r(f) - n_i^l(f) = n_{i+1}^r(f) - n_{i+1}^l(f) + 1 \Leftrightarrow$$
$$\Phi(e_{i-1}, g_i) = \Phi(e_i, g_{i+1}) + 1 \Leftrightarrow$$
$$\Phi(e_{i-1}, g_i) + \Phi(g_i^*) = \Phi(e_i, g_{i+1}) + \Phi(e_i^*).$$

Thus, the flow conservation holds.

Finally, we have $\Phi(e_{\text{len}(f)}, g_{\text{len}(f)}) = 1$, because otherwise the number of edges with slope 1 on the left and the right path differs. Thus the flow conservation holds for $l[e_{\text{len}(f)}^*]$, too.

Now, within an inner face the flow conservations are satisfied. Next, we consider an edge e with its incident faces f_1 and f_2. Edge e has the same slope whether you look at it from f_1 or f_2 and thus the duel arc e^* has the same flow in both faces, too. Thus, the flow is well-defined.

Last, we need to give the arcs flow 1, that go from the source to an edge with slope 1. Arcs from an edge with slope 1 to the sink also get flow 1. The other arcs have flow 0. Thus we have constructed a valid flow of $\mathcal{N}_G$.

"⇒" For the other direction let Φ be a valid flow of $\mathcal{N}_G$. Let Γ be the by Φ induced drawing of (G, ℓ). Now we need to show, that Γ is an LP2-drawing.

Like in Theorem 3.3 we show that the three invariants I.1, I.2 and I.3 hold for every level and prove it from the lowest to the highest level. Let l be a level of G and the invariants be true for all levels below l. Invariants I.1 and I.2 are true because again we only consider *st*-graphs. The proof is the same as in Theorem 3.3.

Now we prove the third invariant I.3, that for every distinct vertices v and w on the same level, v and w have distinct x-coordinates. We distinguish the two cases if v and w share an inner face or not.

For Case 1 v and w share a common inner face f. Then Invariant I.3 is true, because we have an arc in Φ that is in f and cuts level l. Thus by Lemma 4.3 and Lemma 4.1 the distance between v and w is greater than 0.

For Case 2 v and w lie on the same level l but have no common inner face. Without loss of generality in the ordering τ, that is given by the embedding, $\tau(v) < \tau(w)$. Then there are vertices x and y on l with $\tau(x) = \tau(v) + 1$ and $\tau(y) = \tau(w) - 1$. Vertices v and x a common inner face and thus by Case 1 the distance between v and x is greater than 0. The same is true for y and w. If $x = y$ or if x and y share an inner face, then the distance between x and y is at least 0. Then the distance between v and w is greater than 0. Otherwise x and y do not share an inner face. Then we can make the same procedure that we have done for v and w now for x and y. We repeat this recursively until some vertices share the same inner face. Then the distance between v and w is greater than 0, too.

Thus in each case the distance between distinct vertices on the same level is greater than 0 and therefore Invariant I.3 holds. Because we used only slopes -1 and 1 in the construction of Γ, we know that Γ is an LP2-drawing.

□

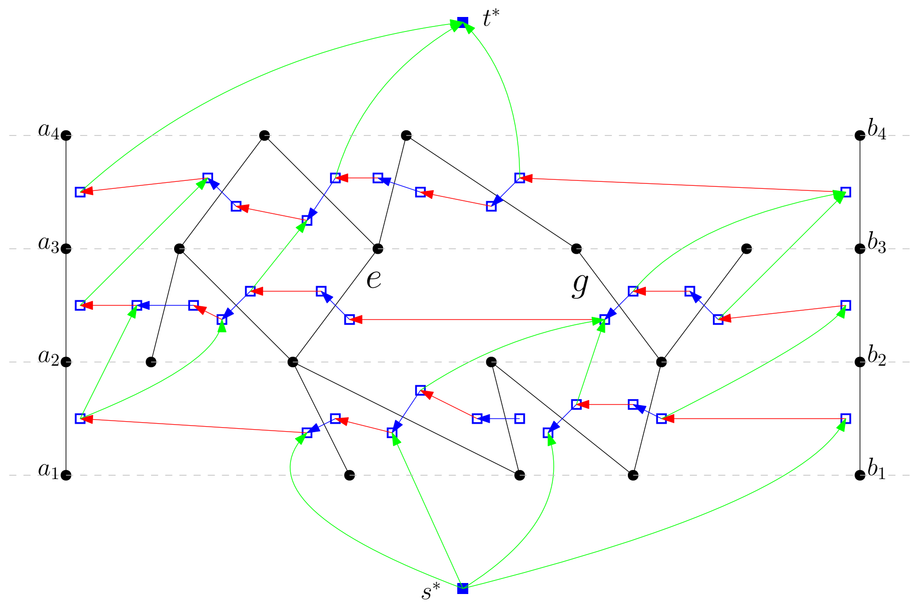


Figure 4.3: A graph with left and right boundary and corresponding flow network

4.2 Flow Network for LP2-Drawings

In the previous section we described an algorithm that can, given an level planar st-graph, decide if there is an LP2-drawing of the graph. In this section we present a similar algorithm, that can operate on every level planar graph. We assume that the graph is connected. Otherwise we apply the algorithm to each component individually.

By expanding the problem, we lose some restriction we could make before. But some conditions like the following, clearly still are satisfied in an LP2-drawing.

Fork: For every vertex u with two neighbors v and w on the same level, the slopes of uv and uw are fixed. Thus, for instance if v lies left of w and both v and w are on the level above u, then uv has slope -1 and uw slope 1.

The algorithm of this section is again based on a flow network. However, before constructing the flow network we transform the level plane graph (G, ℓ) to a level plane graph (G', ℓ') by adding vertices and edges to G. So $G \subset G'$ and for every vertex $x \in G$ we have $\ell'(x) = \ell(x)$ and $\tau'(x) = \tau(x)$. Let $l_1, l_2, \ldots, l_n$ be the levels of G that contain vertices, with l_1 the lowest and l_n the highest level. For every level l_i with $i \in \{1, \ldots, n\}$ we add one vertex a_i with $\ell'(a_i) = l_i$ and $\tau'(a_i) < \tau'(x)$ for every vertex x with $\ell'(x) = l_i$. Similarly, for every level l_i with $i \in \{1, \ldots, n\}$ we add one vertex b_i with $\ell'(b_i) = l_i$ and $\tau'(b_i) > \tau'(x)$ for every vertex x with $\ell'(x) = l_i$. Next we add edges $a_i a_{i+1}$ as *left boundary* and edges $b_i b_{i+1}$ as *right boundary* for $i \in \{1, \ldots, n-1\}$. For the following paragraphs we refer to (G', ℓ') just by (G, ℓ). So we assume that G has left and right boundary edges as described above. In Figure 4.3 the black graph is such a graph with left and right boundary. The added vertices are explicitly labeled.

For an edge $e = xy$ let $\text{right}(e) = uv$ be the edge with $e \neq uv$, $\ell(x) = \ell(u)$, $\tau(u) \geq \tau(x)$, $\ell(y) = \ell(v)$, $\tau(v) \geq \tau(y)$, and where $\tau(u) + \tau(v)$ is minimal. We call this edge *right edge of* e. Intuitively it is the nearest edge of e to the right between the same two levels. Similarly let $\text{left}(e) = wz$ be the edge with $e \neq wz$, $\ell(x) = \ell(w)$, $\tau(w) \leq \tau(x)$, $\ell(y) = \ell(z)$, $\tau(z) \leq \tau(y)$, and where $\tau(w) + \tau(z)$ is maximal. This edge will be called *left edge of* e and is intuitively the nearest edge of e to the left between the same two levels. In Figure 4.3 the edge g is the right edge of e (and also e is the left edge of g). Because (G, ℓ) is planar, this edges are

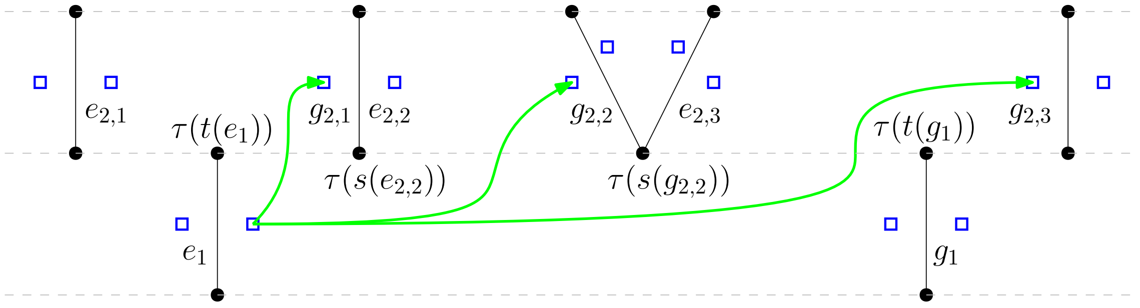


Figure 4.4: An example for placing the green arcs

unique. Edges that are on the left boundary only have a right edge but no left edge and edges on the right boundary only have a left edge but no right edge.

In Section 4.1 we define the width of an *st*-face on one level l as the distance between the two vertices on l. Here the width of a face at l cannot be defined that way, because f could have more then two vertices at l. Therefore we introduce some new notations. For each edge e we refer to the lower vertex (upper vertex, respectively) of e as the *source of the edge*, $s(e)$, (*sink of the edge*, $t(e)$, respectively). Now for two edges e, g connecting the same two levels we define the *distance of the sinks of the edges* $\Delta_{up}(e, g)$ and the *distance of the sources of the edges* $\Delta_{low}(e, g)$.

The definition of a flow network $\mathcal{N} = (N, A)$ is the same as in Section 4.1. The construction of the network $\mathcal{N}_G$ corresponding to a given level plane graph (G, ℓ) with a left and right boundary is slightly different. An example is shown in Figure 4.3. First we add a dual arc e^* for each edge e in G that is not a left or right boundary edge. Those are the blue edges in Figure 4.3. Those arcs go from a *right node* $r[e^*]$ to a *left node* $l[e^*]$. As before, we identify the flow Φ over those arcs with the slope of the inducing edge e: $\Phi(e^*) = 1$ ($\Phi(e^*) = 0$, respectively) if and only if e has slope 1 (-1, respectively). We set the lower bound and capacity for those edges as $\lambda(u, v) = 0$ and $\mu(u, v) = 1$. A left boundary edge has only the right node and a right boundary edge has only the left node.

Next we add arcs between a left node of an edge g_1 to a right node of an edge e_1, where e_1 is the left edge of g_1, so $\text{left}(g_1) = e_1$. These are the red arcs is Figure 4.3. For the sake of brevity, we refer to the edges by $(g_1, e_1) = (l[g_1], r[e_1])$. We set $\lambda(g_1, e_1) = 1$ and $\mu(g_1, e_1) = \infty$.

Next let e_1 be an edge that is not a right boundary edge and let be $g_1 = \text{right}(e_1)$. We look at the edges that are one level above e_1 and g_1. The only edges that we consider are edges g_2 with $\tau(s(g_2)) > \tau(t(e_1))$ and where $e_2 = \text{left}(g_2)$ has $\tau(s(e_2)) < \tau(t(g_1))$ and where $s(e_2) \neq s(g_2)$. An example of three such edges denoted as $g_{2,1}$, $g_{2,2}$ and $g_{2,3}$ is given in Figure 4.4. Now we add arcs $(r[e_1], l[g_2])$ from e_1 to every such edge g_2. These arcs are the green ones in Figure 4.3. We refer to those arcs by (e_1, g_2) and give them $\lambda(e_1, g_2) = 1$ and $\mu(e_1, g_2) = \infty$, too. Note that an arc of one of the last two types always lies in exactly one face. In Figure 4.5 a drawing of one face with the corresponding flow is pictured.

At last we add a source node s^* with arcs $(s^*, l[g])$ for every edge g that is between the levels l_1 and l_2 and where $s(g) \neq s(e)$ for $e = \text{left}(g)$. Additionally we add a sink node t^* with arcs $(r[e], t^*)$ for every edge e that is between the levels l_{n-1} and l_n and where $t(e) \neq t(g)$ for $g = \text{right}(e)$. Those arcs (u, v) get $\lambda(u, v) = 1$ and $\mu(u, v) = \infty$.

For a flow Φ of $\mathcal{N}_G$, let $\Phi_{out}(u_e)$ denote the *outgoing flow of a node* u_e corresponding to an edge e without the flow through e^*. So $\Phi_{out}(u_e) = \sum_{(u_e,v) \in A \setminus \{e^*\}} \Phi(u_e, v)$. Let $\Phi_{in}(v_e)$ be the *incoming flow of a node* v_e corresponding to the edge e without the flow that goes over e^*. Then $\Phi_{in}(v_e) = \sum_{(u,v_e) \in A \setminus \{e^*\}} \Phi(u, v_e)$. Note that for an edge e and its right edge

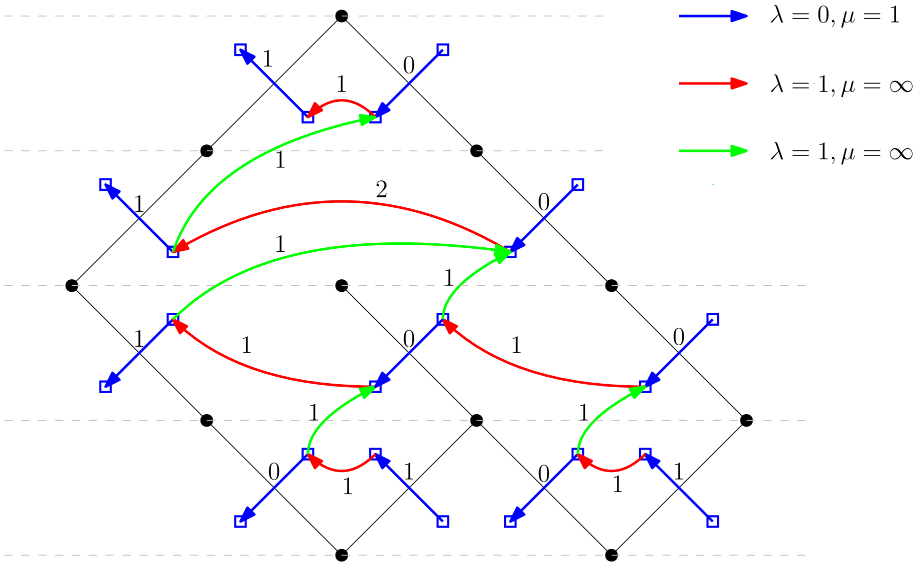


Figure 4.5: One face with network arcs

$g = \text{right}(e)$ we have $\Phi_{out}(r[e]) = 0$ ($\Phi_{in}(l[g]) = 0$, respectively) if and only if $t(e) = t(g)$ ($s(e) = s(g)$, respectively).

Next let (G, ℓ) be a level planar graph with a left and right boundary. We show, that for a flow of $\mathcal{N}_G$ we can construct an LP2-drawing of (G, ℓ) and also that every LP2-drawing where the boundary edges have slope -1 can be described by a flow of $\mathcal{N}_G$. For this we first show a minor statement that will be used later in the proof.

Lemma 4.6. *Let (G, ℓ) be a level plane graph with a left and right boundary and $\mathcal{N}$ the corresponding network. Let Φ be a valid flow of $\mathcal{N}$ and let e and $g = \text{right}(e)$ be two edges of G. Assume a drawing Γ of (G, ℓ) that identifies the slopes and dual arcs as usual. If we have $2 \cdot \Phi_{in}(l[g^*]) = \Delta_{low}(e, g)$ for Γ then also $2 \cdot \Phi_{out}(r[e^*]) = \Delta_{up}(e, g)$.*

Proof. Let (G, ℓ), $\mathcal{N}$ and Φ be as stated above. Additionally let e be an edge of G and $g = \text{right}(e)$. Assume a drawing Γ where the coordinates of $s(e)$ and $s(g)$ are fixed such that $2 \cdot \Phi_{in}(l[g^*]) = \Delta_{low}(e, g)$. Then the coordinates of $t(e)$ and $t(g)$ are fixed by the slopes and we have:

$$t(e) = s(e) + \text{slope}(e)$$
$$t(g) = s(g) + \text{slope}(g).$$

Note that $s(g) - s(e) = \Delta_{low}(e, g)$ and thus $\Delta_{up}(e, g) = \Delta_{low}(e, g) - \text{slope}(e) + \text{slope}(g)$. Furthermore observe that we have $\Phi_{out}(l[g^*]) = \Phi_{in}(r[e^*])$ for every such pair e, g. We can now make a case analysis on the flows over e^* and g^* to proof the lemma. The case distinction is also pictured in Figure 4.6.

Case 1: $\Phi(e^*) = 0$ and $\Phi(g^*) = 0$: Then

$$\Phi_{out}(r[e^*]) = \Phi_{in}(r[e^*]) = \Phi_{out}(l[g^*]) = \Phi_{in}(l[g^*]).$$

Additionally $\Delta_{up}(e, g) = \Delta_{low}(e, g)$, because the edges e and g have the same slope. Therefore we have $2 \cdot \Phi_{out}(r[e^*]) = \Delta_{up}(e, g)$.

Case 2: $\Phi(e^*) = 0$ and $\Phi(g^*) = 1$: Then we have

$$\Phi_{out}(r[e^*]) = \Phi_{in}(r[e^*]) = \Phi_{out}(l[g^*]) = \Phi_{in}(l[g^*]) + 1.$$

Also $\Delta_{up} = \Delta_{low} + 2$ because $\text{slope}(e) =$ -1 and $\text{slope}(g) = 1$. Then $2 \cdot \Phi_{out}(r[e^*]) = 2 \cdot \Phi_{in}(l[g^*]) + 2 = \Delta_{low}(e, g) + 2 = \Delta_{up}(e, g)$.

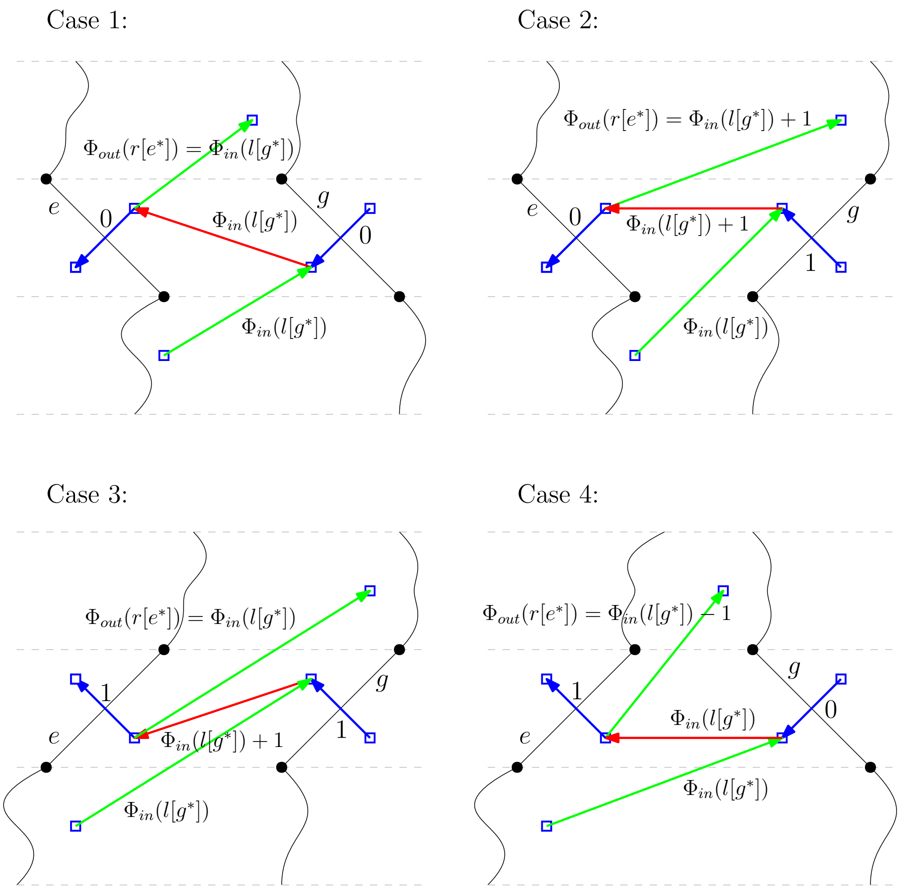


Figure 4.6: Case analysis for flows over e^* and g^* to prove correlation of flows and distances

Case 3: $\Phi(e^*) = 1$ and $\Phi(g^*) = 1$: In this case we have

$$\Phi_{out}(r[e^*]) = \Phi_{in}(r[e^*]) - 1 = \Phi_{out}(l[g^*]) - 1 = \Phi_{in}(l[g^*]).$$

Similarly to the first case, $\Delta_{up}(e, g) = \Delta_{low}(e, g)$, because again slope$(e)$ = slope(g). Then we have $2 \cdot \Phi_{out}(r[e^*]) = \Delta_{up}(e, g)$.

Case 4: $\Phi(e^*) = 1$ and $\Phi(g^*) = 0$: Similar to the second case we have

$$\Phi_{out}(r[e^*]) = \Phi_{in}(r[e^*]) - 1 = \Phi_{out}(l[g^*]) - 1 = \Phi_{in}(l[g^*]) - 1.$$

Because of the flow of the dual arcs we have slope(e) = 1 and slope(g) = -1 and thus $\Delta_{up}(e, g) = \Delta_{low}(e, g) - 2$. Then we see that for the last case $2 \cdot \Phi_{out}(r[e^*]) = 2 \cdot \Phi_{in}(l[g^*]) - 2 = \Delta_{low}(e, g) - 2 = \Delta_{up}(e, g)$.

□

Now we prove the duality between a valid flow and an LP2-drawing as requested.

Lemma 4.7. *Let (G, ℓ) be a level plane graph with a left and a right boundary and $\mathcal{N}_G$ the resulting network. Every flow Φ of $\mathcal{N}_G$ corresponds bijectively to an LP2-drawing Γ of (G, ℓ) with slope -1 for all boundary edges.*

Proof. Let (G, ℓ) be a level plane graph with a left and a right boundary. Furthermore let $\mathcal{N}_G$ be the corresponding network.

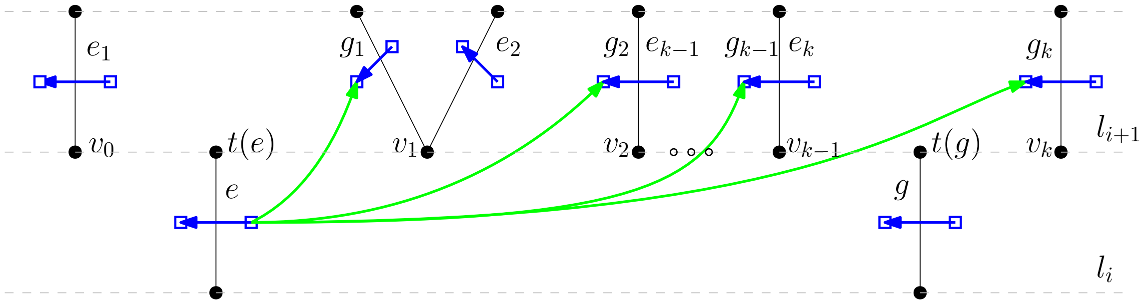


Figure 4.7: Example of a graph G at level l_{i+1}

"$\Rightarrow$ " Let Φ be a valid flow of $\mathcal{N}_G$. We give a schema how to construct an LP2-drawing of (G, ℓ) from it. First we construct an LP2-drawing of (G, ℓ).

For this we start at level l_1. We chose the x-coordinates of the leftmost vertex, that is a_1, arbitrarily. Now we place the other vertices on l_1. For this let be $v \in l_1 \setminus \{a_1\}$ and let g be the leftmost edge that is incident to v. Out of construction this edge has an arc $(s^*, l[g^*])$. Consider the vertex u with $\tau(v) = \tau(u) + 1$. Then we choose the x-coordinates x_v as $x_v = x_u + 2 \cdot \Phi(s^*, l[g^*])$, where x_u is the x-coordinate of u. Since $\Phi(s^*, l[g^*]) > 0$ we do not place two distinct vertices at the same coordinates. For $e = \text{left}(g)$ we see that e is incident to u because G is connected. Thus $2 \cdot \Phi_{in}(l[g^*]) = \Delta_{low}(e, g)$.

Next, we assume that we already have placed the vertices up to level l_i with $i < n-2$ such that

C.1 The drawing of G up to level l_i is an LP2-drawing and

C.2 For every two edges e and $g = \text{right}(e)$ between level l_i and l_{i+1} we have $2 \cdot \Phi_{in}(l[g^*]) = \Delta_{low}(e, g)$.

Now we want to place the vertices on level l_{i+1} in such a way that Condition C.1 and Condition C.2 are satisfied for l_{i+1}, too. For this we first assign the slopes to the edges that are between level l_i and l_{i+1}. If d is an edge between level l_i and l_{i+1} we set

$$\text{slope}(d) = \begin{cases} \text{-1} & \text{if } \Phi(d^*) = 0 \text{ or } d \in \{a_i a_{i+1}, b_i b_{i+1}\} \\ 1 & \text{if } \Phi(d^*) = 1 \end{cases}$$

For the vertices on level l_{i+1} that have neighbors on level l_i the slopes of the connecting edges imply their coordinates. Because Condition C.2 holds for level l_i and by Lemma 4.6 we have $2 \cdot \Phi_{out}(r[e^*]) = \Delta_{up}(e, g)$ for all edges $e \neq b_i b_{i+1}$ between the levels l_i and l_{i+1} and $g = \text{right}(e)$. Thus the sinks of e and g are assigned to distinct coordinates if $t(e) \neq t(g)$, because we have $\Phi_{out}(r[e^*]) > 0$.

There may be vertices on level l_{i+1} that do not have neighbors on l_i. In the following we assign coordinates to them. Let e and g be edges between l_i and l_{i+1} with $g = \text{right}(e)$. An example of that scenario is pictured in Figure 4.7. Let $v_1, \ldots, v_{k-1}$ be vertices between $t(e)$ and $t(g)$ in the order they are embedded on l_{i+1}. Those vertices cannot be incident to edges between the levels l_i and l_{i+1}, because otherwise g would not be the right edge of e. So because G is connected we know that they are connected to edges between the levels l_{i+1} and l_{i+2}. Let g_j denote the leftmost edge with $s(g_j) = v_j$ for $j \in \{1, \ldots, k-1\}$. Those edges have an arc (e, g_j). We refer to the left edge of g_j as $e_j = \text{left}(g_j)$. Observe that every edge g_j has such a left edge, because $\tau(t(e)) < \tau(v_j)$ and thus $g_j \neq a_i a_{i+1}$. Moreover observe that the source of e_j is always $s(e_j) = v_{j-1}$ for $1 < j < k$. For $j = 1$ denote the source of e_j with v_0.

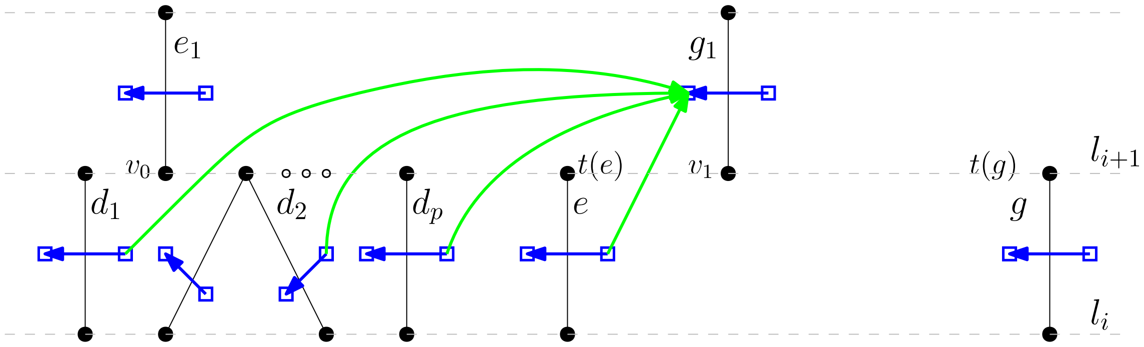


Figure 4.8: Example of a graph G at level l_{i+1} for placing v_1

Denote by e_k the right most edge incident to v_{k-1}, by $g_k = \text{right}(e_k)$ the right edge of e_k and with v_k its source. Either $\tau(v_k) = \tau(t(g))$ or $\tau(v_k) > \tau(t(g))$, but always $\tau(v_k) > \tau(t(e))$ holds. Also $\tau(v_{k-1}) < \tau(t(g))$ and thus by construction of $\mathcal{N}_G$ there exists an arc (e, g_k). With that we have denoted all outgoing arcs from e.

Now, while placing the remaining vertices, we show that Condition C.2 holds for level l_{i+1}. By the definition of the incoming flow of a node and Lemma 4.6 we have $2 \cdot \sum_{j=1}^{k} \Phi(e, g_j) = 2 \cdot \Phi_{out}(r[e^*]) = \Delta_{up}(e, g)$. For $1 < j < k$ we place the x-coordinates at $x_{v_j} = x_{t(e)} + 2 \cdot \sum_{m=1}^{j} \Phi(e, g_m)$. We have $\Phi_{in}(r[g_j^*]) = \Phi(e, g_j)$ for $1 < j < k$, because for the source v_{j-1} of the left edge of g_j we have $\tau(t(e)) < \tau(v_{j-1})$ and $\tau(v_j) < \tau(t(g))$ and thus there is no other arc going to g_j. Thus, for every edge g_j with $1 < j < k$ we have $2 \cdot \Phi_{in}(r[g_j^*]) = \Delta_{low}(e_j, g_j)$.

For $j = k$ we distinguish two cases. If $\tau(v_k) = \tau(t(g))$ then also $v_k = t(g)$ and the x-coordinate of v_k is already set. Then again $2 \cdot \Phi_{in}(r[g_k^*]) = \Delta_{low}(e_k, g_k)$. Otherwise, we have $\tau(v_k) > \tau(t(g))$. Then there exists an edge d with upper vertex $t(d) = t(g)$ and (d, g_k). We place v_k depending on d, because $\tau(v_k) > \tau(t(d))$. Note that the distance of $\max\{t(e), v_{k-1}\}$ and $t(g)$ is $2 \cdot \Phi(e, g_k)$.

For $j = 1$ and $k \neq 1$ we set $x_{v_1} = x_{up(e)} + 2 \cdot \Phi(e, g_1)$. For $j = 1 = k$ the previous case can be applied. To show that $2 \cdot \Phi_{in}(r[g_1^*]) = \Delta_{low}(e_1, g_1)$ holds, we make a case analysis.

Case 1: $\tau(v_0) = \tau(t(e))$**:** Then $\Phi_{in}(r[g_1^*]) = \Phi(e, g_1)$ and thus $2 \cdot \Phi_{in}(r[g_1^*]) = 2 \cdot \Phi(e, g_1) = \Delta_{low}(e_1, g_1)$.

Case 2: $\tau(v_0) < \tau(t(e))$**:** An example of this case is shown in Figure 4.8. We have edges d_1 to d_p between the levels l_i and l_{i+1} that are left of e and have an arc (d_q, g_1) for $q \in \{1, \ldots, p\}$. Because $\tau(t(e)) < \tau(v_1)$, the arc (d_q, g_1) is the last arc for the edge d_q. It follows that $2 \cdot \sum_{q=1}^{p} \Phi(d_q, g_1) = x_{t(e)} - x_{v_0}$. Hence, we have $2 \cdot \Phi(e, g_1) = x_{v_1} - x_{t(e)}$. Then it holds $2 \cdot \Phi_{in}(r[g_1^*]) = \Delta_{low}(e_1, g_1)$.

So we have proved Condition C.2. With this procedure we place the vertices at level l_{i+1} in the same order as in the embedding and always place them to distinct coordinates if they are disjunct. Thus the drawing of (G, ℓ) up to level l_{i+1} is an LP2-drawing and so Condition C.1 holds, too.

After placing all vertices up to level l_{n-1}, we only need to place the vertices at l_n. Let e be an edge between the levels l_{n-1} and l_n. If $x_{s(e)}$ is the x-coordinate of the source of e, then we assign the x-coordinate of the sink of e to $x_{t(e)} = x_{s(e)} + \text{slope}(e)$. For any two edges e and $g = \text{right}(e)$ between the levels l_{n-1} and l_n we have $2 \cdot \Phi_{out}(r[e^*]) = \Delta_{up}(e, g)$. We have $\Phi_{out}(r[e^*]) = 0$ if and only if the sinks of e and g are the same vertex. Thus this placing of the vertices results into an LP2-drawing of

(G, ℓ), because the vertices are sill assigned to distinct coordinates if they are disjunct. There is no other vertex on l_n except for the sinks of some edge d between the levels l_{n-1} and l_n, because G is connected. That way we have constructed an LP2-drawing of (G, ℓ) from a flow Φ of $\mathcal{N}_G$.

"$\Leftarrow$" Now we have an LP2-drawing Γ of (G, ℓ) and find a valid flow over $\mathcal{N}_G$. For all arcs $(s^*, l[g^*])$ we set $\Phi(s^*, l[g^*]) = \frac{\Delta_{low}(\text{left}(g), g)}{2}$. Because we only placed arcs when $\Delta_{low}(\text{left}(g), g) > 0$, we know that $\Phi(s^*, l[g^*]) \geq \lambda(s^*, l[g^*]) = 1$. Moreover we have $2 \cdot \Phi_{in}(l[g^*]) = \Delta_{low}(\text{left}(g), g)$.

Now let level l_i be a level of (G, ℓ) with $i < n - 1$. We next consider all arcs that go from a node corresponding to an edge e with e between the levels l_{i-1} and l_i or below. As an induction hypothesis assume that we have assigned flows to all these arcs such that from s^* to them Φ is a valid flow. Furthermore assume that for every edge $g \neq a_i a_{i+1}$ between the levels l_i and l_{i+1} we have $2 \cdot \Phi_{in}(l[g^*]) = \Delta_{low}(\text{left}(g), g)$. For each edge d between l_i and l_{i+1} we identify the flow value of the dual arc d^* with the slope of d as usual. Now we have assigned the incoming flow of each left node corresponding to an edge between the now considered two levels l_i and l_{i+1}.

The only outgoing arc for one of these left nodes $l[g^*]$ is to the right nodes $r[e^*]$ of the left edge of g, so $e = \text{left}(g)$. We assign the incoming flow $\Phi_{in}(l[g^*])$ plus the flow through g^* to the arc (g, e). That means we have $\Phi(g, e) = \Phi_{in}(l[g^*]) + \Phi(g^*)$. Thus the flow conservation at $l[g^*]$ is fulfilled.

Next, let e be an edge between the levels l_i and l_{i+1} and let $g = \text{right}(e)$. The only flow that still needs to be assigned for e is the flow over arcs (e, g') with g' an edge between the levels l_{i+1} and l_{i+2}. So let g' be an edge between l_{i+1} and l_{i+2} where an arc (e, g') exists. We set

$$\Phi(e, g') = \frac{\min(s(g'), t(g)) - \max(s(\text{left}(g')), t(e))}{2}.$$

Again we have only placed arcs when $\Delta_{low}(\text{left}(g'), g') > 0$ and $\Delta_{up}(e, g) > 0$. Also $\tau(s(\text{left}(g'))) < \tau(t(g))$ and $\tau(s(g')) > \tau(t(e))$. Thus we know that $\Phi(e, g') \geq \lambda(e, g') = 1$.

In the following we prove the flow conservation at $r[e^*]$. By induction hypothesis, we know that $2 \cdot \Phi_{in}(l[g^*]) = \Delta_{low}(e, g)$. By a similar claim as made in Lemma 4.6 we see that $2 \cdot \Phi_{out}(r[e^*]) = \Delta_{up}(e, g)$. However, $\Phi_{out}(r[e^*])$ is the sum of flow values $\Phi(e, g')$. Additionally $\Delta_{up}(e, g)$ can be displayed as sum over the distances

$$\min(low(g'), up(g)) - \max(low(\text{left}(g')), up(e))$$

for edges g' with an arc (e, g'). So the flow conservation at $r[e^*]$ is satisfied.

Now we have assigned flow to every arc of $\mathcal{N}_G$ except of the arcs (g, e) and arcs d^* with g, e, d between the levels l_{n-1} and l_n and arcs $(r[e^*], t^*)$. For the first two types of arcs, (g, e) and d^*, we do the same as for the other levels, so the flow conservation at $l[g^*]$ is satisfied. Now we only need the flow through arcs $(r[e^*], t^*)$. For this we assign the flow value in such a way that the flow conservation at $r[e^*]$ is satisfied. We know that $\Phi(r[e^*], t^*) \geq \lambda(r[e^*], t^*) = 1$ because again we only have arcs $(r[e^*], t^*)$ if $\Delta_{up}(e, \text{right}(e)) > 0$ and this distance corresponds directly to the flow that needs to go out of $r[e^*]$.

Since the flow conservation is fulfilled at every node we have $\Phi_{out}(s^*) = \Phi_{in}(t^*)$. Therefore, the constructed flow Φ is a valid flow of $\mathcal{N}_G$.

□

Note that if (G, ℓ) has no left and no right boundary we can always extend it to a graph (G', ℓ') that has one. An LP2-drawing of (G, ℓ) can then be easily constructed from an LP2-drawing of (G', ℓ'). Also if we have an LP2-drawing of (G, ℓ), it is easy to extend it to an LP2-drawing of (G', ℓ'), where all boundary edges have slope -1. Furthermore note that even if the LP2-drawing of the graph (G', ℓ') is restricted to having slope -1 for the boundary edges, the LP2-drawing of the primary graph (G, ℓ) is not further restricted.

Algorithm 4.1 summarized the results above and given a level plane graph decides whether it has an LP2-drawing.

Algorithm 4.1: Algorithm to construct an LP2-drawing of a level plane graph

Input : level plane graph (G, ℓ)
Output : LP2-drawing of (G, ℓ)

1. Extend (G, ℓ) to (G', ℓ') by adding boundary edges.
2. Construct $\mathcal{N}_{G'}$ out of (G', ℓ').
3. Check whether $\mathcal{N}_{G'}$ has a valid flow Φ.
4. Construct an LP2-drawing Γ' of (G', ℓ') out of Φ as described in Lemma 4.7.
5. Delete the previously added boundary edges from Γ' to get an LP2-drawing of (G, ℓ).

Next we take a look at the running time of Algorithm 4.1. For this let $|V|$ be the number of vertices of G. Step 1. and Step 5. can be done in $O(|V|)$, because we need to add or delete two vertices for each level, but we can only have as many levels as we have vertices. Step 2. can be done in $O(|V|)$, too. To construct $\mathcal{N}_{G'}$ we need to add an arc for each edge. For this note that the number of edges in G is in $O(|V|)$, because the degree of each vertex of G is limited to 4. Therefore the number of nodes in $\mathcal{N}_{G'}$ is in $O(|V|)$, too. But since the resulting network is planar out of construction, the number of arcs in $\mathcal{N}_{G'}$ is also in $O(|V|)$. To find those arcs in constant time $O(1)$ we can use the ordering τ of the vertices on the levels given by the plane embedding. To find the flow Φ in Step 3. there exist many algorithms. Note that our network is st-planar and that we are only interested in determining whether our network has a valid flow. Furthermore we can choose positive arc costs. On the other hand our flow is bounded not only by the capacity but also by a lower bound. Tamassia and Garg [GT97] presented a minimum cost flow algorithm that can find such a flow in $O(|V|^{7/4}\sqrt{\log|V|})$ and is often used in similar drawing algorithms. This algorithm can be used here, too, and additionally gives us the smallest drawing of (G', ℓ'). At last Step 4. can be done in $O(|V|)$, because we need to set the coordinates of each vertex of G' exactly once. These coordinates only depend on one previous vertex on the same level or the slopes of the (at most 4) incident edges.

Hence, in summary the run time of Algorithm 4.1 only depends on the run time $T(|V|)$ of finding the flow in the network. This results in the following theorem.

Theorem 4.8. *Let (G, ℓ) be a level plane graph. An LP2-drawing of (G, ℓ) can be constructed in $O(|V|^{7/4}\sqrt{\log|V|})$ time by Algorithm 4.1, if there exists one.*

5. Conclusion

In this thesis we have proposed several ways for constructing drawings of level planar graphs with fixed slopes. For this we defined level planar drawings with the two slopes -1 and 1, which we referred to as LP2-drawings. Those drawings we restricted further to rectangular LP2-drawings, where each inner face is a rectangle. We have shown that we can construct a rectangular LP2-drawing of a level plane *st*-graph in polynomial time, if such a drawing exists. For this we presented a 2-SAT-based algorithm.

For LP2-drawings without restriction on the shape of the faces we first presented an algorithm that produces such drawings for level planar *st*-graphs if there exists one. Afterwards we extended that algorithm to work on every level planar graph. This algorithm first creates a flow network out of the graph and then transforms a valid flow of that network into an LP2-drawing.

Next, we will give an outlook on problems that could be of further interest.

In Chapter 4 we have given a running time for Algorithm 4.1. This running time depends on the time $T(|V|)$ that is needed to be invested in finding a flow Φ. It is likely that the given run time $T(|V|) \in O(|V|^{7/4}\sqrt{\log|V|})$ can be further improved.

In this thesis we only considered LP2-drawings, but the algorithm we used could be extended to create level planar drawings that use n different slopes. For this consider a set $\mathcal{S}$ of n slopes, that are equidistant and denote them by s_0 to s_{n-1}, where $s_0 < s_1 < \cdots < s_{n-1}$. We call drawings that are level planar and only use slopes out of $\mathcal{S}$ LP-$\mathcal{S}$-drawings. We still want to use the flow network idea, to construct an LP-$\mathcal{S}$-drawing out of a level plane graph. The construction of the network stays the same as in Chapter 4.2, but we have to adjust the capacities. We set the capacity of every dual arc e^* of an edge e to $\mu(e^*) = n-1$. For constructing the drawing of the graph, we now identify the slope of an edge e by the values of the flow through the dual arc e^*:

$$\text{slope}(e) = s_{\Phi(e^*)}.$$

For the left and right boundary of the extended graph (G', ℓ') we always choose slope s_0.

To proof that this network can be used as requested it seems advisable to first restrict the slope set $\mathcal{S}$ so that the distance between the upper vertices of two edges with slope s_i and s_{i+1} is 2, if the distance between the lower vertices is 0 for $i \in \{0, \ldots, n-2\}$. Because the distance between two adjacent levels is 1 that means $s_{i+1} - s_i = 2$ or in general $s_i - s_j = 2 \cdot (i - j)$. Then the proof that a valid flow Φ of $\mathcal{N}$ corresponds bijectively to an

LP-$\mathcal{S}$-drawing of (G, ℓ) is the same as in Lemma 4.7 with one exception. Lemma 4.6 needs to be adjust to the n different slopes. For other equidistant slope sets the proof can be adjusted easily.

It can also be considered if it is possible to adjust the flow-based algorithms in such a way that non-equidistant slope sets are possible.

Another interesting question would be to find the minimum number of slopes that is necessary to draw a level planar graph. Both equidistant and non-equidistant slope sets could be considered.

Our results provide implications on partial settings of the LP2-drawings. Let G' be the subgraph of G, where the LP2-drawing Γ' of G' is already provided as a part of the input. For the algorithm based on 2-SAT we add the clause $s(f, e) \leftrightarrow \text{slope}(e)$ for each edge $e \in G'$ incident to a face f with a slope slope(e) in Γ'. If a satisfying assignment ϕ of $\mathcal{S}(G, \ell)$ plus the new clauses exists, then it can be transformed into a rectangular LP2-drawing as described above. For the algorithms based on network flows we have to choose the lower bound and the capacity for some edges differently. If we have an edge $e \in G'$ with slope(e) in Γ' and the dual arc e^* in the network $\mathcal{N}$, we set

$$\lambda(e^*) = \mu(e^*) = \begin{cases} 0 & \text{slope}(e) = \text{-}1 \\ 1 & \text{slope}(e) = 1. \end{cases}$$

Thus we fix the flow over the dual arcs. With both adjustments by fixing the slopes the new resulting partial drawing of G' is equivalent to Γ' up to translation along the x-axis, since Γ' is connected.

For rectangular LP2-drawings an equivalent extension of the 2-SAT based algorithm can be done to find partial *simultaneous* drawings of two graphs. For this we denote with (G_1, ℓ_1) the first and with (G_2, ℓ_2) the second graph. Let G' be a subgraph of both G_1 and G_2. We want that the drawing Γ'_1 of G' as part of the drawing Γ_1 is equivalent to the drawing Γ'_2 of G' as part of the drawing Γ_2. To do that we combine the two 2-SAT instances $\mathcal{S}(G_1, \ell_1)$ and $\mathcal{S}(G_2, \ell_2)$ and clauses as follows. For each edge $e_1 \in G_1$ that is in G' and its in G' equivalent edge $e_2 \in G_2$ we add the clause $s(f_1, e_1) \leftrightarrow s(f_2, e_2)$ for some faces f_1 and f_2 incident to e_1 and e_2, respectively. If a satisfying assignment ϕ of this new 2-SAT instance exists, then it can be transformed into two rectangular LP2-drawings Γ_1 and Γ_2 of (G_1, ℓ_1) and (G_2, ℓ_2), similar to Theorem 3.3. For the part of G_1 and G_2 that should be drawn simultaneously we have $\Gamma'_1 = \Gamma'_2$, again up to translation along the x-axis.

Also the problem of drawing parts of two graphs simultaneously for non-rectangular LP2-drawings or even level planar drawings with more than two slopes could be looked at.

Bibliography

[BMW06] János Barát, Jiří Matoušek, and David R. Woodf. Bounded-degree graphs have arbitrarily large geometric thickness. *the electronic journal of combinatorics*, 13(R3):1–14, 2006.

[BT88] Giuseppe Di Battista and Roberto Tamassia. Algorithms for plane representations of acyclic digraphs. *Theoretical Computer Science*, 61(2):175 – 198, 1988.

[CPR90] Jurek Czyzowicz, Andrzej Pelc, and Ivan Rival. Drawing orders with few slopes. *Discrete Mathematics*, 82(3):233 – 250, 1990.

[DESW07] Vida Dujmović, David Eppstein, Matthew Suderman, and David R. Wood. Drawings of planar graphs with few slopes and segments. *Computational Geometry*, 38(3):194 – 212, 2007.

[DSW07] Vida Dujmović, Matthew Suderman, and David R. Wood. Graph drawings with few slopes. *Computational Geometry*, 38(3):181 – 193, 2007.

[ELT96] PETER EADES, XUEMIN LIN, and ROBERTO TAMASSIA. An algorithm for drawing a hierarchical graph. *International Journal of Computational Geometry & Applications*, 06(02):145–155, 1996.

[F4́8] István Fáry. On straight line representation of planar graphs. *Acta Univ. Szeged. Sect. Sci. Math.*, 11:229–233, 1948.

[FRA08] FABRIZIO FRATI. On minimum area planar upward drawings of directed trees and other families of directed acyclic graphs. *International Journal of Computational Geometry & Applications*, 18(03):251–271, 2008.

[GT95] Ashim Garg and Roberto Tamassia. Upward planarity testing. *Order*, 12(2):109–133, Jun 1995.

[GT97] Ashim Garg and Roberto Tamassia. A new minimum cost flow algorithm with applications to graph drawing. In Stephen North, editor, *Graph Drawing*, pages 201–216, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

[HK04] Patrick Healy and Ago Kuusik. Algorithms for multi-level graph planarity testing and layout. *Theoretical Computer Science*, 320(2):331 – 344, 2004.

[HN13] Patrick Healy and Nikola Nikolov. *Hierarchical Drawing Algorithms.* 08 2013.

[KPP10] Balázs Keszegh, János Pach, and Dömötör Pálvölgyi. Drawing planar graphs of bounded degree with few slopes. In *Graph Drawing*, 2010.

[KPPT08] Balázs Keszegh, János Pach, Dömötör Pálvölgyi, and Géza Tóth. Drawing cubic graphs with at most five slopes. *Computational Geometry*, 40(2):138 – 147, 2008.

[PT95] Achilleas Papakostas and Ioannis G. Tollis. Improved algorithms and bounds for orthogonal drawings. In Roberto Tamassia and Ioannis G. Tollis, editors, *Graph Drawing*, pages 40–51, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.

[Riv93] Ivan Rival. *Reading, Drawing, and Order*, pages 359–404. Springer Netherlands, Dordrecht, 1993.

[STT81] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(2):109–125, Feb 1981.

[Tam87] Roberto Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.*, 16:421–444, 1987.

[TTV91] Roberto Tamassia, Ioannis G. Tollis, and Jeffrey Scott Vitter. Lower bounds for planar orthogonal drawings of graphs. *Inf. Process. Lett.*, 39(1):35–40, 1991.