AN SPQR-TREE-LIKE EMBEDDING REPRESENTATION FOR LEVEL PLANARITY[*†]

Guido Brückner[‡] *and Ignaz Rutter*[§]

Abstract. An SPQR-tree is a data structure that efficiently represents all planar embeddings of a biconnected planar graph. It is a key tool in a number of constrained planarity testing algorithms, which seek a planar embedding of a graph subject to some given set of constraints.

We develop an SPQR-tree-like data structure that represents all level-planar embeddings of a biconnected level graph with a single source, called the LP-tree, and give an algorithm to compute it in linear time. Moreover, we show that LP-trees can be used to adapt three constrained planarity algorithms to the level-planar case by using LP-trees as a drop-in replacement for SPQR-trees.

1 Introduction

Testing planarity of a graph and finding a planar embedding, if one exists, are classical algorithmic problems. For visualization purposes, it is often desirable to draw a graph subject to certain additional constraints, e.g., finding orthogonal drawings [41] or symmetric drawings [32], or inserting an edge into an embedding so that few edge crossings are caused [29]. Historically, these problems have been considered for embedded graphs. More recent research has attempted to optimize not only one fixed embedding, but instead to optimize over all possible planar embeddings of a graph. This includes (i) orthogonal drawings [11], (ii) simultaneous embeddings, where one seeks to embed two planar graphs that share a common subgraph such that they induce the same embedding on the shared subgraph (see [10, 40] for a survey), (iii) simultaneous orthogonal drawings [3], (iv) embeddings where some edge intersections are allowed [1], (v) inserting an edge [29], a vertex [15], or multiple edges [16] into an embedding, (vi) partial embeddings, where one insists that the embedding extends a given embedding of a subgraph [4], and (vii) finding minimum-depth embeddings [6, 7].

The common tool in all of these recent algorithms is the SPQR-tree data structure, which efficiently represents all planar embeddings of a biconnected planar graph G by breaking down the complicated task of choosing a planar embedding of G into the task of independently choosing a planar embedding for each triconnected component of G [21, 22,

*This work was funded by the German Research Foundation (DFG) under grant RU 1901/3-1.

†A preliminary version of this work appeared as G. Brückner, I. Rutter, *An SPQR-Tree Like Embedding Representation of Level Planarity*, ISAAC'20, 8:1–15, 2020.

‡*Karlsruhe Institute of Technology (KIT)*, guido.brueckner@kit.edu

§*University of Passau*, rutter@fim.uni-passau.de

23, 33, 38, 42]. This is a much simpler task since the triconnected components have a very restricted structure, and so the components offer only basic, well-structured choices.

An *upward planar drawing* is a planar drawing where each edge is represented by a y-monotone curve. For a *level graph* $G = (V, E)$, which is a directed graph where each vertex $v \in V$ is assigned to a level $\ell(v)$ such that $\ell(u) < \ell(v)$ for each edge $(u, v) \in E$, a *level-planar drawing* is an upward planar drawing where each vertex v is mapped to a point on the horizontal line $y = \ell(v)$. Level planarity can be tested in linear time [24, 35, 36, 39]. Recently, the problem of extending partial embeddings for level-planar drawings has been studied [13]. While the problem is NP-hard in general, it can be solved in polynomial time for single-source graphs. Very recently, an SPQR-tree-like embedding representation for upward planarity has been used to extend partial upward embeddings [12], see also [17]. Its construction crucially relies on an existing decomposition result for upward-planar graphs [34]. No such result exists for level-planar graphs. Moreover, the level assignment leads to components of different "heights", which makes our decompositions significantly more involved.

Contribution. We develop the LP-tree, an analogue of SPQR-trees for level-planar embeddings of level graphs with a single source whose underlying undirected graph is biconnected. It represents the choice of a level-planar embedding of a level-planar graph by individual embedding choices for certain components of the graph, for each of which the embedding is either unique up to reflection, or allows to arbitrarily permute certain subgraphs around two pole vertices. Its construction is based on suitably modifying the SPQR-tree of G, which represents all planar embeddings of G, not just the level-planar ones, such that, eventually, the modified tree represents exactly the level-planar drawings of G. See Figure 1 (a, b) for examples of how level planarity is more restrictive than planarity. The size of the LP-tree is linear in the size of G and it can be computed in linear time. The LP-tree is a useful tool that unlocks the large amount of SPQR-tree-based algorithmic knowledge for easy translation to the level-planar setting. In particular, we obtain linear-time algorithms for partial and constrained level planarity for biconnected single-source level graphs, which improves upon the $O(n^2)$-time algorithm known to date [13]. Further, we describe the first efficient algorithm for the simultaneous level planarity problem when the shared graph is a biconnected single-source level graph.

We first introduce important concepts and notation that we use throughout the paper in Section 2. We show the existence of LP-trees in Section 3. The proof is constructive and immediately gives a polynomial-time algorithm, which we then improve to run in linear time. In Section 4, we present three applications of LP-trees. Finally, we give some concluding remarks in Section 5.

2 Preliminaries

Let $G = (V, E)$ be a connected level graph. We assume further that for each vertex $v \in V$, we are given a value $d(v) \geq \ell(v)$ called the *demand* of v. Demands provide an interface to model the restrictions imposed on the embeddings of one biconnected component by other biconnected components; see Figure 1 (c).

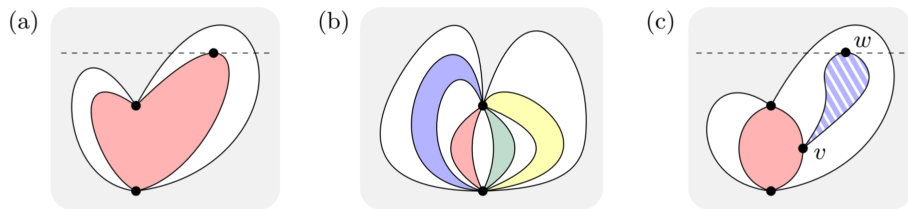

Figure 1: In (a), the height of the red component makes it impossible to flip it. In (b), note that the red and green components can be exchanged, as can the blue and yellow components, but neither the blue nor the yellow component can be embedded between the red and green component. In (c), set the demand of v as $d(v) = \ell(w)$ in the LP-tree that represents the graph that consists of the red and gray part (but not the striped blue part). This models the restriction imposed on the embedding of the red subgraph by the striped blue biconnected component.

An *apex* of some vertex set $V' \subseteq V$ is a vertex $v \in V'$ whose level is maximum. We write $\text{apex}(V')$ for the set of all apices of V'. The *demand* of V', denoted by $d(V') = \max\{d(v) \mid v \in V'\}$, is the maximum demand of a vertex in V'. Similarly, we write $\ell(V') = \max\{\ell(v) \mid v \in V'\}$ to denote the maximum level of the vertices in V'. An apex of a face f is an apex of the vertices incident to f, and we denote the set of all apices of f by $\text{apex}(f)$.

A *planar drawing* of G is a planar drawing of the underlying undirected graph of G in the plane. Planar drawings are *equivalent* if they can be continuously transformed into each other without creating intermediate intersections. A *planar embedding* is an equivalence class of equivalent planar drawings. Planar embeddings of connected graphs are usually represented by specifying a *rotation system*, which defines the clockwise cyclic order of the edges around each vertex, and an outer face.

Level Graphs and Level-Planar Embeddings. A *path* is a sequence $(v_1, v_2, \dots, v_j)$ of vertices so that for $1 \leq i < j$ either $(v_i, v_{i+1}) \in E$ or $(v_{i+1}, v_i) \in E$. A *directed path* is a sequence $(v_1, v_2, \dots, v_j)$ of vertices so that $(v_i, v_{i+1}) \in E$ for $1 \leq i < j$. A vertex u *dominates* a vertex v if there exists a directed path from u to v. A vertex is a *sink* if it dominates no vertex except for itself. A vertex is a *source* if it is dominated by no vertex except for itself. An *st-graph* is a graph with a single source and a single sink, usually denoted by s and t, respectively.

For the remainder of this paper we restrict our considerations to level-planar drawings of G where each vertex $v \in V$ that is not incident to the outer face is incident to some inner face f so that each apex a of f satisfies $d(v) < \ell(a)$. We call such drawings *level-planar with demand d*, and say that G is *level-planar with demand d*, if it admits a corresponding drawing. Note that setting $d(v) = \ell(v)$ for each $v \in V$ gives the conventional definition of level-planar drawings.

We sketch in Section 4 how to use demands to restrict the admissible embeddings of biconnected components in the presence of cutvertices. A planar embedding Γ of G is *level-planar (with demand d)* if there exists a level-planar drawing of G (with demand d)

whose underlying planar embedding is Γ. We then call Γ a *level-planar embedding (with demand d)*.

To simplify the exposition, we preprocess our input level graph $G = (V, E)$ on k levels to a level graph G' on $d(V) + 1$ levels as follows. We obtain G' from G by adding a new vertex t on level $d(V) + 1$ with demand $d(t) = d(V) + 1$, connecting it to all vertices on level k and adding the edge (s, t). Note that G' is generally not an st-graph. Let H be a graph with a level-planar embedding Γ and let H' be a supergraph of H with a level-planar embedding Γ'. The embedding Γ' *extends* Γ if Γ' and Γ coincide on H. The embeddings of G' where the edge (s, t) is incident to the outer face and the embeddings of G are, in a sense, equivalent.

Lemma 1. *A planar embedding Γ of G is level-planar if and only if there exists a level-planar embedding Γ' of G' that extends Γ.*

Proof. Let Γ be a planar embedding of G so that there exists a level-planar embedding Γ' of G' that extends Γ. Because G is a subgraph of G', the restriction of Γ' to G, which is Γ, is also level-planar.

Conversely, let Γ be a level-planar embedding of G. Since all apices of V lie on the outer face, the newly added vertex t can be connected to them without causing any edge crossings. Then, because s is the single source of G and t is the sole apex of $V(G')$, the edge (s, t) can be drawn into the outer face as a y-monotone curve without causing edge crossings. Let Γ' refer to the resulting embedding. Then Γ' is a level-planar embedding of G' that extends Γ. □

To represent all level-planar embeddings of G, it is sufficient to represent all level-planar embeddings of G' and to remove t and its incident edges from all embeddings. It is easily observed that if G is a biconnected single-source graph, then so is G'. We assume from now on that the vertex set of our input graph G has a unique apex t and that G contains the edge (s, t). We still refer to the highest level as level k, i.e., the apex t lies on level k.

Level-planar embeddings Γ of a graph G can be characterized by the existence of a supergraph H of G that is an st-graph and that has a planar embedding whose restriction to G is Γ. We use Lemmas 2 and 3, and a novel characterization of single-source level planarity in Lemma 4 to prove that certain planar embeddings are also level-planar.

Lemma 2. *Let $G = (V, E)$ be a level graph with a single-source s and a unique apex t. Further, let Γ be a level-planar embedding of G. Then there exists an st-graph $G_{st} = (V, E \cup E_{st})$ together with a level-planar embedding Γ_{st} that extends Γ.*

Proof. We prove the claim by induction over the number of sinks in G. Note that because t is an apex of G, it must be a sink. So G has at least one sink. If G has one sink, the claim is trivially true for $E_{st} = \emptyset$. Now suppose that G has more than one sink. Let $w \neq t$ be a sink of G. In some level-planar drawing of G with embedding Γ, walk up vertically from w into the incident face above w. If a vertex v or an edge (u, v) is encountered, set $E_{st} = \{(w, v)\}$. If no vertex or edge is encountered, w lies on the outer face of Γ. Then set $E_{st} = \{(w, t)\}$. Note that in both cases the added edge can be inserted into the drawing as a y-monotone curves while maintaining level-planarity, and this decreases the number of sinks by 1. Then extend E_{st} inductively, which shows the claim. □

Next we establish a characterization of the planar embeddings that are level-planar. The following lemma is implicit in the planarity test for st-graphs by Chiba [14] and the work on upward planarity by Di Battista and Tamassia [20].

Lemma 3. *Let G be an st-graph. Then each planar embedding Γ of G where (s,t) is incident to the outer face is also a level-planar embedding of G and vice, versa.*

Proof. Consider a vertex $v \neq s,t$ of G. Then the incoming and outgoing edges appear consecutively around v in Γ. To see this, suppose that there are four vertices $w,x,y,z \in V$ with edges $(w,v),(v,x),(y,v),(v,z) \in E$ that appear in that counter-clockwise cyclic order around v in Γ; see Fig. 2 (a). Because G is an st-graph there are directed paths p_w and p_y from s to w and y, respectively, and directed paths p_x and p_z from x and z to t, respectively. Moreover, $p \in \{p_w,p_y\}$ and $p' \in \{p_x,p_z\}$ are disjoint and do not contain v. Then some $p \in \{p_w,p_y\}$ and $p' \in \{p_x,p_z\}$ must intersect, a contradiction to the fact that Γ is planar.

Let $e_1,e_2,\dots,e_i,e_{i+1},\dots,e_n$ be the counter-clockwise cyclic order of edges around v in Γ so that $e_1,\dots,e_i$ are incoming edges and $e_{i+1},\dots,e_n$ are outgoing edges. In other words, $e_1,\dots,e_i$ denote the left-to-right order of incoming edges and $e_n,e_{n-1},\dots,e_{i+1}$ denote the left-to-right order of outgoing edges. Split the clockwise cyclic order of edges around s at (s,t) to obtain the left-to-right order of outgoing edges. Symmetrically, split the counter-clockwise order of edges around t at (s,t) to obtain the left-to-right order of incoming edges.

Create a level-planar drawing Γ' of G step by step as follows; see Fig. 2. Draw vertices s and t on levels $\ell(s)$ and $\ell(t)$, respectively, and connect them by a straight-line segment. Call the vertices s,t and the edge (s,t) *discovered*. Call the path s,t the *right frontier*. Call a vertex on the right frontier *settled* if all of its outgoing edges are discovered.

More generally, let $s=u_1,u_2,\dots,u_n=t$ denote the right frontier. Modify the right frontier while maintaining that (i) the right frontier is a directed path from s to t, (ii) the edges on the boundary of and in the interior of the cycle formed by the right frontier and the edge (s,t) are precisely the discovered edges, and (iii) the edge (s,t) and the right frontier bound the left and right side of the outer face of Γ', respectively.

Let u_i denote the vertex on the right frontier closest to t that is not settled. Discover the leftmost undiscovered outgoing edges starting from u_i to construct a directed path $v_1 = u_i,v_2,\dots,v_m$, where v_m is the first vertex that had been discovered before. Because G has a single sink such a vertex exists. Because Γ is planar, v_m lies on the right frontier, i.e., $v_m = u_j$ for some j with $i < j \leq n$. Draw the vertices $v_2,\dots,v_{m-1}$ and the edges (v_a,v_{a+1}) for $1 \leq a < m$ to the right of the path $u_i,\dots,u_j$ in Γ' (Property (iii) of the invariant), maintaining level-planarity of Γ'. This creates a new face f of Γ' whose boundary is $u_i,u_{i+1},\dots,u_j = v_m,v_{m-1},\dots,v_1 = u_i$.

We show that f is a face of Γ. Because u_a is settled for $a > i$, there cannot be an undiscovered outgoing edge between (u_{a-1},u_a) and (u_a,u_{a+1}) in the counter-clockwise order of edges around u_a in Γ for $i < a < j$ (see edge g in Fig. 2 (b)). There can also not be a discovered outgoing edge because of Property (ii) of the invariant (see edge e in Fig. 2 (b)).

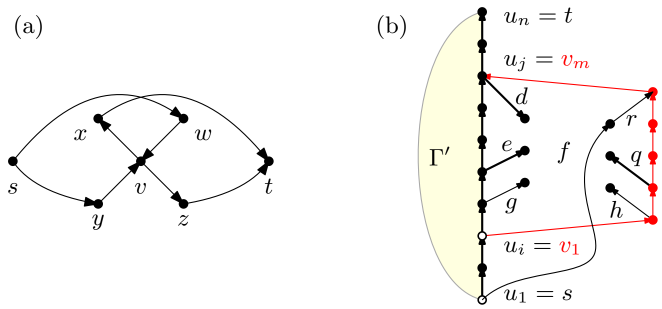

Figure 2: Proof of Lemma 3. The incoming and outgoing edges around each vertex are consecutive (a). Creating the level-planar embedding Γ' by attaching the path $v_1, v_2, \dots, v_m$ (drawn in red) to the right frontier $u_1, u_2, \dots, u_n$, thereby creating a new face f. Discovered edges are drawn thickly. The edges e, g, h, q, r, d cannot exist.

Since we always choose leftmost undiscovered edges, there is no undiscovered outgoing edge between (v_a, v_{a+1}) and (v_{a-1}, v_a) in the counter-clockwise order of edges around v_a in Γ for $1 < a < m$ (see edge h in Fig. 2 (b)). There can also not be a discovered outgoing edge because v_a was not discovered before (see edge q in Fig. 2 (b)). There can be no outgoing edge between (v_1, v_2) and (u_i, u_{i+1}) in the counter-clockwise order of edges around $v_1 = u_i$ because either such an edge would be discovered contradicting Property (ii), or not, contradicting the fact that (v_1, v_2) is chosen as the leftmost undiscovered outgoing edge of v_1. There can be no outgoing edge between (u_{j-1}, u_j) and (v_{m-1}, v_m) in the counter-clockwise order of edges around $u_j = v_m$ because either $u_j = v_m = t$ is a sink, or the incoming and outgoing edges appear consecutively around $u_j = v_m$ in Γ (see edge d in Fig. 2 (b)).

There can also be no incoming edge (u, v) between any of these edge pairs (see edge r in Fig. 2 (b)). This is because G has a single source s, so there exists a directed path p from s to u. Because u lies inside of f, the path p must contain a vertex x on the boundary of f. Then p would also contain an outgoing edge of x which we have just shown to be impossible.

Let $s = u_1, u_2, \dots, u_i = v_1, v_2, \dots, v_m = u_j, \dots, u_n = t$ denote the new right frontier. Note that the invariant holds for this modified right frontier. Because G has a single source, the above procedure discovers (and hence draws) all vertices and edges of G in this way. Because Γ and Γ' have the same faces they are the same embedding. Finally, Γ' is level-planar by construction, which shows the claim. □

Thus, a planar embedding Γ of a graph G is level-planar if and only if it can be augmented to an st-graph $G' \supseteq G$ such that all augmentation edges can be embedded in the faces of Γ without crossings. This gives rise to the following characterization.

Lemma 4. *Let G be a single-source k-level graph with a unique apex t. A planar embedding Γ of G is level-planar if and only if each vertex of v with $\ell(v) < k$ is incident to at least one face f such that v is not an apex of f.*

Proof. Let Γ_l be a level-planar drawing of G. Consider a vertex v with $\ell(v) < \ell(t)$. If v has an outgoing edge (v, w), then v and w are incident to some shared face f. Since $\ell(v) < \ell(w)$, vertex v is not an apex of f. If v has no outgoing edges, start walking upwards from v in a straight line. Stop walking upwards if an edge (u, w) or a vertex w is encountered. Then v and w are again incident to some shared face f. Moreover, $\ell(v) < \ell(w)$, and therefore v is not an apex of f. If no edge or vertex is encountered when walking upwards, v must lie on the outer face. Because t lies on the outer face and $\ell(v) < \ell(t)$, vertex v is not an apex of the outer face. Finally, because Γ_l is level-planar it is, of course, also planar.

Conversely, let Γ_p be a planar embedding of G where every vertex v with $\ell(v) < \ell(t)$ is incident to at least one face of which it is not an apex. The idea is to augment G and Γ_p by inserting edges so that G becomes an st-graph together with a planar embedding Γ_p. To that end, fix for each face f an arbitary apex $a_f \in \text{apex}(f)$ and consider each sink $v \neq t$ of G. By assumption, v is incident to at least one face f so that v is not an apex of f, and hence $\ell(v) < \ell(a_f)$. So the augmentation edge $e = (v, a_f)$ can be inserted into G without creating a cycle. Further, e can be embedded into f.

Because all augmentation edges that are embedded into the same face f have the same endpoint a_f, the embedding Γ_p of G remains planar. This means that G can be augmented so that t becomes the only sink while maintaining the planarity of Γ_p. Because G also has a single source, G is now an st-graph and it follows from Lemma 3 that Γ_p is not only planar, but also level-planar. □

In particular, since all the demands satsisfy $d(v) \geq \ell(v)$ for all $v \in V$, we have the following corollary.

Corollary 1. *Let G be a single-source k-level graph with a unique apex t and let Γ be a planar embedding of G. Then Γ is level-planar with demand d if and only if every vertex v that does not lie on the outer face is incident to a face with an apex a that satisfies $d(v) < \ell(a)$.*

Decomposition Trees and SPQR-Trees. Our description of decomposition trees follows Angelini et al. [2]. Let G be a biconnected graph. A *separation* of G consists of two subgraphs H_1, H_2 of G with $H_1 \cup H_2 = G$ and $H_1 \cap H_2 = \{u, v\}$. Define the tree $\mathcal{T}$ that consists of two *nodes* μ_1 and μ_2 connected by an undirected *arc* as follows. For $i = 1, 2$ node μ_i is equipped with a multigraph $\text{skel}(\mu_i) = H_i + e_i$, called its *skeleton*, where $e_i = (u, v)$ is called a *virtual* edge. The arc (μ_1, μ_2) links the two virtual edges e_i in $\text{skel}(\mu_i)$ with each other. We also say that the virtual edge e_1 *corresponds* to μ_2 and likewise that e_2 corresponds to μ_1. The idea is that $\text{skel}(\mu_1)$ provides a more abstract view of G where e_1 serves as a placeholder for H_2. More generally, there is a bijection $\text{corr}_\mu \colon E(\text{skel}(\mu)) \to N(\mu)$ that maps every virtual edge of $\text{skel}(\mu)$ to a neighbor of μ in $\mathcal{T}$, and vice versa. For an arc (ν, μ) of $\mathcal{T}$, the virtual edges e_1, e_2 with $\text{corr}_\mu(e_1) = \nu$ and $\text{corr}_\nu(e_2) = \mu$ are called *twins*, and e_1 is called the *twin* of e_2 and vice versa. This procedure is called a *decomposition*; see Fig. 3 on the left. It can be re-applied to skeletons of the nodes of $\mathcal{T}$, which leads to larger trees with smaller skeletons. A tree obtained in this way is a *decomposition tree* of G. A decomposition can be undone by *contracting* an arc (μ_1, μ_2) of $\mathcal{T}$, forming a new node μ with a larger skeleton as follows. Let e_1, e_2 be twin edges in $\text{skel}(\mu_1), \text{skel}(\mu_2)$. The skeleton of μ is the union of $\text{skel}(\mu_1)$ and $\text{skel}(\mu_2)$ without the two twin edges e_1, e_2. Contracting all

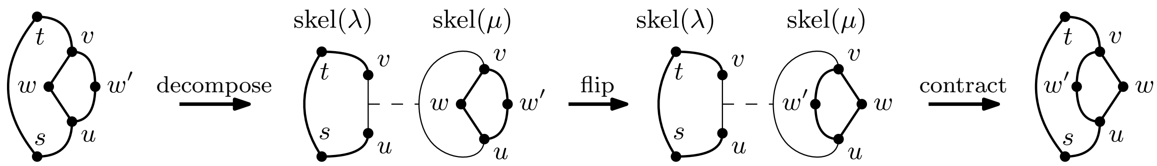

Figure 3: Decompose the embedded graph G on the left at the separation pair u, v. This gives the center-left decomposition tree whose skeletons are embedded as well. Reflecting the embedding of skel(μ) or, equivalently, flipping (λ, μ), yields the same decomposition tree with a different embedding of skel(μ). Contract (λ, μ) to obtain the embedding on the right.

arcs of a decomposition tree of G results in a decomposition tree consisting of a single node whose skeleton is G; see Fig. 3 on the right. Let μ be a node of a decomposition tree with a virtual edge e with $\text{corr}_\mu(e) = \nu$. The *expansion graph* of e and ν in μ, denoted by $G(e)$ and $G(\mu, \nu)$, respectively, is the graph obtained by removing the twin of e from skel(ν) and contracting all arcs in the subtree that contains ν.

Each skeleton of a decomposition tree of G is a minor of G. So if G is planar, each skeleton of a decomposition tree $\mathcal{T}$ of G is planar as well. If (μ_1, μ_2) is an arc of $\mathcal{T}$, and skel(μ_1) and skel(μ_2) have fixed planar embeddings Γ_1 and Γ_2, respectively, then the skeleton of the node μ obtained from contracting (μ_1, μ_2) can be equipped with an embedding Γ by merging these embeddings along the twin edges corresponding to (μ_1, μ_2); see Fig. 3 on the right. This requires at least one of the virtual edges e_1 in skel(μ_1) with $\text{corr}_{\mu_1}(e_1) = \mu_2$ or e_2 in skel(μ_2) with $\text{corr}_{\mu_2}(e_2) = \mu_1$ to be incident to the outer face. If we equip every skeleton with a planar embedding and contract all arcs, we obtain a planar embedding of G. This embedding is independent of the order of the edge contractions. Thus, every decomposition tree $\mathcal{T}$ of G *represents* (not necessarily all) planar embeddings of G by choosing a planar embedding of each skeleton and contracting all arcs.

Let e_{ref} be an edge of G, called the *reference edge*. Rooting $\mathcal{T}$ at the unique node μ_{ref} whose skeleton contains the real edge e_{ref} identifies a unique parent virtual edge in each of the remaining nodes; all other virtual edges are called *child virtual edges*. We direct the arcs of $\mathcal{T}$ from the root towards the leaves. This determines for each node $\mu \neq \mu_{\text{ref}}$ a unique *parent virtual edge* in skel(μ). We call its endpoints the *poles* of μ. Restricting the embeddings of the skeletons so that the parent virtual edge (the edge e_{ref} in case of μ_{ref}) is incident to the outer face, we obtain a representation of (not necessarily all) planar embeddings of G where e_{ref} is incident to the outer face. Let μ be a node of $\mathcal{T}$ and let e be a child virtual edge in skel(μ) with $\text{corr}_\mu(e) = \nu$. Then the expansion graph $G(\mu, \nu)$ is simply referred to as $G(\nu)$.

The *SPQR-tree* is a special decomposition tree whose skeletons are precisely the triconnected components of G. It has four types of nodes: S-nodes, whose skeletons are cycles, P-nodes, whose skeletons consist of three or more parallel edges between two vertices, and R-nodes, whose skeletons are simple triconnected graphs. Finally, a Q-node has a skeleton consisting of two vertices connected by one real and by one virtual edge. In the skeletons of all other node types all edges are virtual. Moreover, no two S-nodes and no two P-nodes can be adjacent. In an SPQR-tree the embedding choices are of a particularly

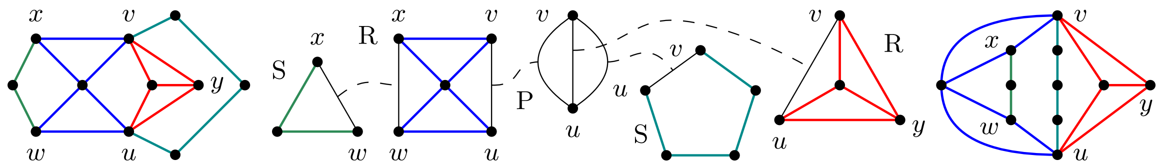

Figure 4: A planar graph on the left and its SPQR-tree in the middle. The five nodes of the SPQR-tree are represented by their respective skeleton graphs. Dashed edges connect twin virtual edges and colored edges correspond to Q-nodes. The embedding of the graph on the right is obtained by flipping the embedding of the blue R-node and swapping the middle and right edge of the P-node.

simple form. The skeletons of Q- and S-nodes have a unique planar embedding (not taking into account the choice of the outer face). The child virtual edges of P-node skeletons may be permuted arbitrarily, and the skeletons of R-nodes are 3-connected, and thus have a unique planar embedding up to reflection [43]. See Fig. 4 and Fig. 7 (a,b) for examples of a planar graph and its SPQR-tree.

3 A Decomposition Tree for Level Planarity

We construct a decomposition tree of a given single-source level graph $G = (V, E)$ whose underlying undirected graph is biconnected that represents all level-planar embeddings of G, called the *LP-tree*. As noted in Section 2, we assume that G has a unique apex t, for which $\ell(t) = \ell(V)$.

The LP-tree for G is constructed based on the SPQR-tree for G. We keep the notion of S-, P-, Q- and R-nodes and construct the LP-tree so that the nodes behave similarly to their namesakes in the SPQR-tree. The skeleton of a P-node consists of two vertices that are connected by at least four parallel virtual edges that can be permuted arbitrarily. The skeleton of an R-node μ is equipped with a *reference embedding* Γ_μ, and the choice of embeddings for such a node is limited to either Γ_μ or its reflection. Unlike in SPQR-trees, the skeleton of μ need not be triconnected, instead it can be an arbitrary biconnected planar graph. We note that, in SPQR-trees, P-nodes have at least three parallel virtual edges. However a P-node that has only three parallel virtual edges has a fixed embedding up to reflection, and we therefore consider such nodes as R-nodes.

In the following we first determine necessary conditions on the embeddings of the skeletons of the SPQR-tree in order to obtain a level-planar embedding. Afterwards, we modify the SPQR-tree so that it only represents embeddings that satisfy these necessary conditions and show that they are also sufficient, i.e., the resulting decomposition tree represents exactly the level-planar embeddings. Finally, we show that the construction can be performed in linear time.

3.1 Necessary Conditions

Let $G = (V, E)$ be a biconnected single-source level graph with demand function d and let $\mathcal{T}$ be its SPQR-tree. As a first step, we study necessary conditions on the embeddings of skeletons of $\mathcal{T}$ for the corresponding embedding of G to be level-planar with demand d. We will frequently use the following criterion for showing that a particular embedding is not level-planar with demand d.

Lemma 5. *Let G be a level graph with a fixed planar embedding Γ, let C be a cycle of G and let v be a vertex that is embedded in the interior of C. If $d(v) \geq \ell(C)$, then Γ is not level-planar with demand d.*

Proof. Let x be a vertex that is embedded in the interior of C and that maximizes $d(x)$ among all such vertices. If Γ is level-planar with demand d, then x must be incident to a face whose apex a satisfies $\ell(a) > d(x)$ by Corollary 1. If a lies in the interior of C, then $d(a) \geq \ell(a) > d(x)$ contradicts the choice of x. If a lies on C, then $\ell(a) > d(x) \geq d(v) \geq \ell(C)$ is a contradiction. □

Let μ be a node of a decomposition tree of a graph G with poles u, v. We define $d(\mu)$ as the maximum level of any vertex in $G(\mu)$, except for u and v. The following lemma gives a necessary condition on the ordering of the children of P-nodes for level-planar embeddings with demand d.

Lemma 6. *Let $\mathcal{T}$ be a decomposition tree of a biconnected single-source graph G that represents a planar embedding Γ of G that is level-planar with demand d. Let further μ be a P-node of $\mathcal{T}$ with poles u, v such that $\ell(u) \leq \ell(v)$.*

Let $\nu_1, \dots, \nu_n$ be the children of μ ordered non-decreasingly according to $d(\nu_j)$ and let ε_j denote the virtual edge of $\mathrm{skel}(\mu)$ that corresponds to ν_j. Let $1 \leq i \leq n$ be the largest index such that $d(\nu_i) < \ell(v)$ if such a component exists, and $i = 1$ otherwise. Then for each $i \leq j \leq n$ the virtual edges $\varepsilon_1, \dots, \varepsilon_j$ appear consecutively around u and v in Γ.

We remark that the statement of the lemma does not hold for $j < i$ since the edges $\varepsilon_1, \dots, \varepsilon_i$ can be permuted arbitrarily.

Proof of Lemma 6. Assume for the sake of contradiction that there is a j with $i \leq j \leq n$ for which $\varepsilon_1, \dots, \varepsilon_j$ are not consecutive. Without loss of generality, we may assume that $j > 1$ is smallest with this property. Then there is a child ν_k, $k > j$ such that ε_k is embedded between ε_1 and ε_j.

Let v_1, v_j, v_k be vertices in $G(\nu_1), G(\nu_j), G(\nu_k)$ with maximum demand, respectively. Let further p_1, p_j, p_k be a simple path from u to v in G_1, G_j, G_k containing v_1, v_j, v_k, respectively. The paths p_1, p_j together form a simple cycle C that contains v_k in its interior. Since $k > j > 1$, we have $d(\nu_k) \geq d(\nu_j), d(\nu_1)$. Moreover $\max\{d(\nu_j), d(\nu_1), \ell(v)\}$ is an upper bound for $\ell(C)$. Thus, C contains in its interior C the vertex v_k with $d(v_k) \geq \ell(C)$. Hence Γ is not level-planar with demand d by Lemma 5. □

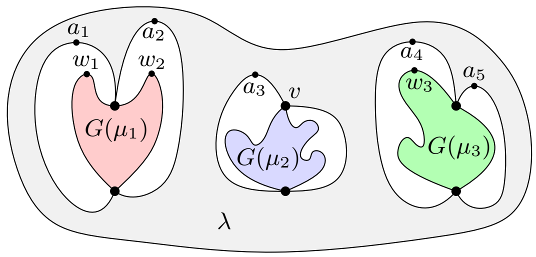

Figure 5: In the figure we assume $d(v) = \ell(v)$ for all vertices. Then $d(\mu_1) = \ell(w_1) = \ell(w_2)$, $d(\mu_2) \leq \ell(v) - 1$ and $d(\mu_3) = \ell(w_3)$. The space around μ_1 is $\ell(a_1)$, the space around μ_2 is $\ell(v)$ and the space around μ_3 is $\ell(a_5)$. Observe that for μ_1 and μ_2 the demand is strictly less than the space around them, and therefore $G(\mu_1)$ and $G(\mu_2)$ can be mirrored without violating level-planarity. On the other hand, $d(\mu_3)$ is greater than the space around μ_3, and mirroring $G(\mu_3)$ results in an embedding that is not level-planar.

Next, we also develop necessary conditions that involve the skeletons of R-nodes. To this end, we assume that the decomposition tree $\mathcal{T}$ represents at least one planar embedding Γ_{ref} that is level-planar (with demand d). We further assume that the reference embeddings of the R-node skeletons are chosen consistently with Γ_{ref}.

Let λ, μ be two R-nodes such that λ is the parent of μ. We call λ and μ *relatively fixed* if in each embedding of G that is represented by $\mathcal{T}$ and that is level-planar with demand d, the skeletons of λ and μ are either both the reference embedding, or they are both the flipped version of the reference embedding. Note that Γ_{ref} guarantees that it is not necessary to flip exactly one of them.

Let ε denote the edge of λ that represents μ and let g_1, g_2 denote the two faces of Γ_{ref} that project to the faces f_1, f_2 left and right of ε in $\text{skel}(\mu)$, respectively. Whether it is possible to flip in Γ_{ref} the embedding of $G(\mu)$ depends on the demand $d(\mu)$ and on the apices of g_1 and g_2. Namely, if both apex sets have a level that is strictly greater than $d(\mu)$, then $G(\mu)$ can be flipped; see the components $G(\mu_1), G(\mu_2)$ in Fig. 5. On the other hand, if the demand $d(\mu)$ exceeds the level of the lower of the two apex sets, i.e., $d(\mu) \geq \min\{\ell(g_1), \ell(g_2)\}$, then $G(\mu)$ cannot be flipped; see the component $G(\mu_3)$ in Fig. 5. Motivated by this, we call $\min\{\ell(g_1), \ell(g_2)\}$ the *space around* ε.

To incorporate this information into our decomposition tree, we associate to each face f of each R-node skeleton μ a value $\text{space}(f)$ as follows. If f is an inner face, take g as the face of Γ_{ref} that projects to f and set $\text{space}(f) = \ell(g)$. If f is the outer face, we set $\text{space}(f) = \infty$. The following lemma shows that the space around edges of R-nodes is the same for all level-planar embeddings with demand d, i.e., the definition of space requires only the existence of some level-planar embedding (with demand d), but is otherwise independent of Γ_{ref}.

Lemma 7. *Let $\mathcal{T}$ be a decomposition tree of a biconnected single-source graph G, let μ be an R-node of $\mathcal{T}$ and let f be an inner face of* $\text{skel}(\mu)$. *Let further Γ, Γ' be embeddings of G that are level-planar with demand d and that are both represented by $\mathcal{T}$ and let g, g' be the faces of Γ and Γ', respectively, that project to f in* $\text{skel}(\mu)$. *Then* $\text{apex}(g) = \text{apex}(g')$.

Proof. Assume for the sake of contradiction that $\mathrm{apex}(g) \neq \mathrm{apex}(g')$. Let $C_g, C_{g'}$ denote the facial cycles of g and g'. After possibly mirroring Γ' (which changes neither the fact that it is level-planar with demand d, nor the apices of g'), we may assume that the skeleton of μ has the same embedding in Γ and Γ'. We may further assume that $\ell(C_g) \geq \ell(C_{g'})$ and if $\ell(C_g) = \ell(C_{g'})$ that further $\mathrm{apex}(C_g) \setminus \mathrm{apex}(C_{g'}) \neq \emptyset$. Note that if $\ell(C_g) > \ell(C_{g'})$, then trivially $\mathrm{apex}(C_g) \cap \mathrm{apex}(C_{g'}) = \emptyset$. In either case, we find that there exists a vertex $x \in \mathrm{apex}(C_g) \setminus \mathrm{apex}(C_{g'})$, which therefore satisfies $d(x) \geq \ell(x) \geq \ell(C_{g'})$.

Since C_g and $C_{g'}$ both project to f in $\mathrm{skel}(\mu)$, it follows that in Γ all vertices of $V(C_g) \setminus V(C_{g'})$ lie in the interior of $C_{g'}$. In particular x lies in the interior of $C_{g'}$. Then Γ is not level-planar with demand d by Lemma 5. □

Our next goal is to show that indeed this definition of space allows to determine whether two adjacent R-nodes of a decomposition tree $\mathcal{T}$ are relatively fixed.

Lemma 8. *Let $\mathcal{T}$ be a decomposition tree of a biconnected single-source graph G that represents some embedding Γ_{ref} that is level-planar with demand d. Let further λ, μ be R-nodes such that λ is the parent of μ, let ε denote the edge of $\mathrm{skel}(\lambda)$ that represents μ, and let f_1, f_2 be the two faces of $\mathrm{skel}(\lambda)$ incident to ε. If $\min\{\mathrm{space}(f_1), \mathrm{space}(f_2)\} \leq d(\mu)$, then μ and λ are relatively fixed.*

Proof. Without loss of generality, assume $\mathrm{space}(f_1) \geq \mathrm{space}(f_2)$. Assume for the sake of contradiction that there exists an embedding Γ' of G that is level-planar with demand d and that is represented by $\mathcal{T}$ in such a way that exactly one of $\mathrm{skel}(\mu)$ and $\mathrm{skel}(\lambda)$ has its reference embedding. After possibly mirroring Γ', we may assume that $\mathrm{skel}(\lambda)$ has the reference embedding. Let g_1, g_2 be the faces of Γ_{ref} that project to f_1, f_2, respectively, and likeswise let g'_1, g'_2 be the faces of Γ' that project to f_1 and f_2, respectively.

By Lemma 7, $\ell(g_1) = \mathrm{space}(f_1) = \ell(g'_1)$ and $\ell(g_2) = \mathrm{space}(f_2) = \ell(g'_2)$. Let x be a vertex of $G(\mu)$ except for its pole vertices that maximizes $d(x)$. By assumption $d(x) \geq \min\{\mathrm{space}(f_1), \mathrm{space}(f_2)\} = \mathrm{space}(f_2)$.

Observe that by Lemma 5, x must lie on the outer face of every embedding of $G(\mu)$ that is level-planar with demand d. Since Γ_{ref} is level-planar with demand d, x is incident to a face whose apex has level greater than $d(x)$, which can hence only be g_1.

Since in Γ' $\mathrm{skel}(\lambda)$ is mirrored with respect to $\mathrm{skel}(\mu)$ and $\mathrm{skel}(\mu)$ is biconnected, it follows that x is incident to g'_2 in Γ'. But then the demand of x is at least as great as the level of all vertices of its incident faces, and Γ' is not level-planar with demand d by Corollary 1. □

3.2 Constructing the LP-tree

The construction of the LP-tree starts out with an SPQR-tree $\mathcal{T}$ of G and uses an arbitary fixed embedding Γ_{ref} that is level-planar with demand d, which we use to define the reference embeddings of the R-node skeletons and to associate the space values to faces of the R-node skeletons.

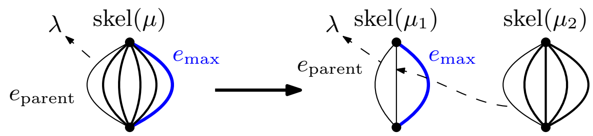

Figure 6: Result of a P-node μ split with parent λ and child with maximum demand ν. Note that after the split, μ_1 is an R-node and μ_2 has one less child than μ had.

At the start, we explicitly label each node of $\mathcal{T}$ as an S-, P-, Q- or R-node. This way, we can continue to talk about S-, P-, Q- and R-nodes of our decomposition tree even when they no longer have their defining properties in the sense of SPQR-trees. As mentioned above, we label P-nodes of $\mathcal{T}$ that have only three virtual edges as R-nodes and equip them with a reference embedding.

Assume the edge (s, t) to be incident to the outer face of every level-planar drawing of G (Lemma 1), i.e., consider $\mathcal{T}$ rooted at the Q-node corresponding to (s, t).

The construction of our decomposition tree works by following the necessary conditions established in Section 3.1. It works in three steps. First, we decompose the tree further by decomposing P-nodes in order to disallow permutations that lead to embeddings that are not level planar (with demand d) according to Lemma 6. Second, we contract all arcs of the decomposition tree that connect an R-node parent to an S-node child. Third, we contract arcs between all pairs of R-nodes that satisfy the condition of Lemma 8 and are therefore relatively fixed. Since all operations either impose necessary conditions on level-planar embeddings (with demand d) or do not change the embeddings that are represented by the tree, the resulting LP-tree represents all embeddings of G that are level-planar with demand d. After giving the details of the construction, we prove that the converse holds as well by showing that every embedding represented by the LP-tree satisfies the conditions of Corollary 1. We start with the first step. Lemma 6 motivates the following modification of a decomposition tree $\mathcal{T}$; see Fig. 6. Take a P-node μ with poles u, v that has a child edge whose demand is at least $\ell(v)$. Denote by λ the parent of μ. Further, let $e_{\max}$ be a child virtual edge with maximum demand and let e_{parent} denote the parent edge of skel(μ). Obtain a new decomposition tree $\mathcal{T}'$ by splitting μ into two nodes μ_1 and μ_2 representing the subgraph H_1 consisting of the edges $e_{\max}$ and e_{parent}, and the subgraph H_2 consisting of the remaining child virtual edges, respectively. Note that the skeleton of μ_1, which corresponds to H_1, has only two child virtual edges. We therefore define it to be an R-node. The other node μ_2 is either a P-node (if it has at least three children) or an R-node (if it has only two children). Observe that in any embedding of skel(μ) that is obtained from choosing embeddings for skel(μ_1) and skel(μ_2) and contracting the arc (μ_1, μ_2), the edge $e_{\max}$ is the first or last child edge. Conversely, since μ_2 allows to arbitarily permute it children (either because it is a P-node or because it is an R-node with just two children) all embeddings where $e_{\max}$ is the first or last child edge are still represented by $\mathcal{T}'$.

For the construction of the LP-tree (see Fig. 7 for an example), we start with the SPQR-tree $\mathcal{S}$ of G and iteratively apply this decomposition, creating new R-nodes on the way, until each P-node μ with poles u and v has only child virtual edges e that have demand at most $\ell(v) - 1$. Denote the resulting decomposition tree by $\mathcal{S}'$; see Fig. 7c.

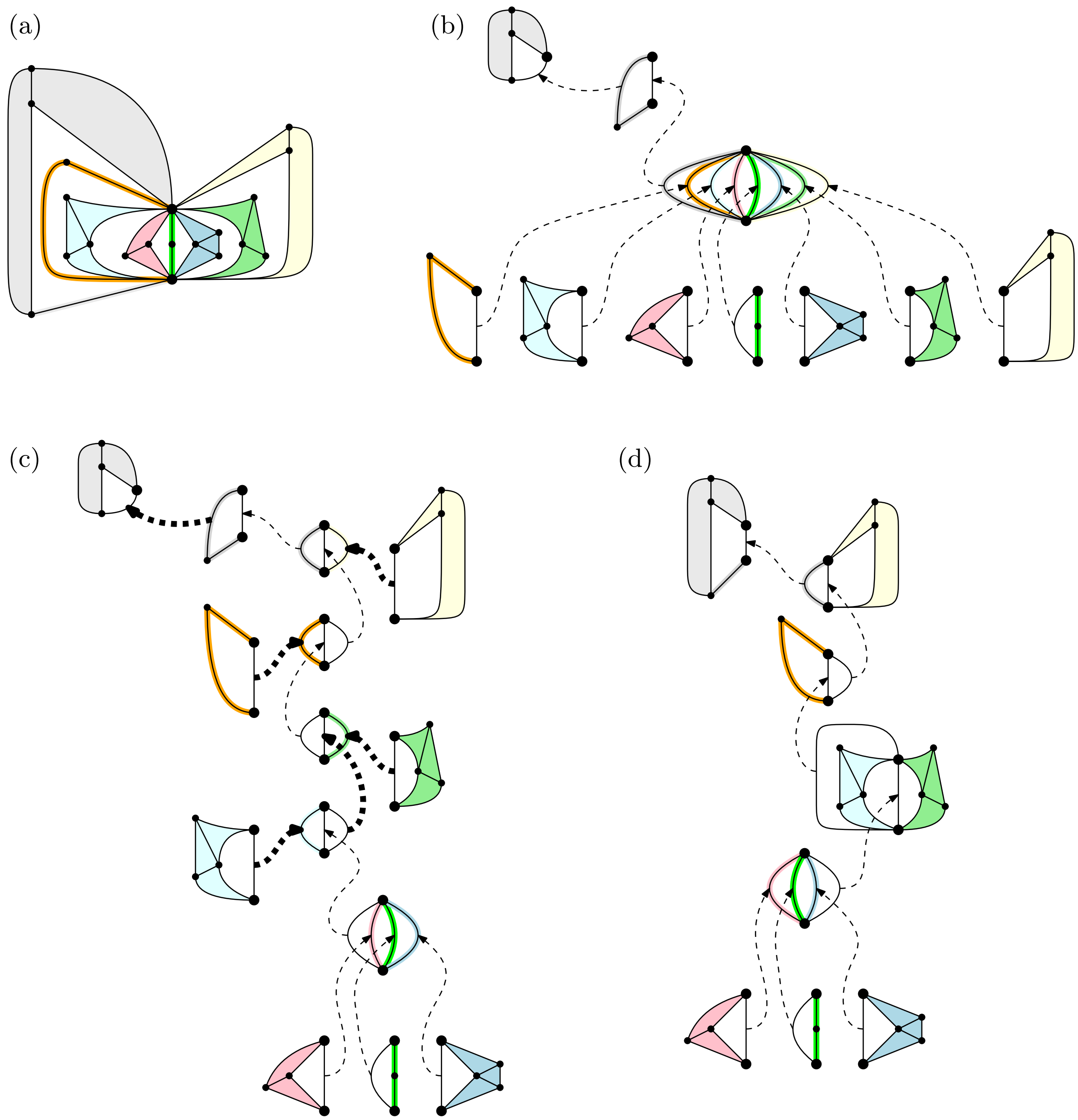

Figure 7: Example construction of the LP-tree for the graph G (a). We start with the SPQR-tree of G (b). Arcs are oriented towards the root. Next, we split the P-node, obtaining the tree shown in (c). Finally, we contract arcs that connect R-nodes with S-nodes and arcs that are between R-nodes that are relatively fixeds (thick dashed lines). This gives the final LP-tree $\mathcal{T}$ for G (d).

In the second step, we modify $\mathcal{S}'$ by contracting all arcs that connect an R-node to a child that is an S-node. Denote the resulting decomposition tree by $\mathcal{S}''$. We equip the resulting R-node with the reference embedding obtained by the contraction of the reference embedding of the R-node parent and the unique embedding of the S-node child.

In the third step, we contract in $\mathcal{S}''$ each arc (λ, μ) between R-nodes μ, λ such that $d(\mu) \geq \min\{\text{space}(f_1), \text{space}(f_2)\}$ where f_1, f_2 are the faces of $\text{skel}(\lambda)$ incident to the edge that represents μ. When performing such a contraction, we equip the resulting R-node with the reference embedding obtained by the contraction of the reference embeddings of $\text{skel}(\lambda)$ and $\text{skel}(\mu)$. The decomposition tree obtained by performing all these contractions is the LP-tree $\mathcal{T}$; see Fig. 7d.

Note that a P-node of $\mathcal{S}$ whose poles u, v satisfy $\ell(u) = \ell(v)$ has $d(\mu) < d(\mu_i)$ for each of its children and it is therefore entirely decomposed into R-nodes with two children, each, in $\mathcal{S}'$. In the third step, these will in fact be contracted into a single R-node (possibly together with some other R-nodes).

By construction, the LP-tree $\mathcal{T}$ has the following properties.

(P1) For every P-node μ with poles u, v such that $\ell(u) \leq \ell(v)$ $d(\mu) < \ell(v)$ holds.

(P2) No R-node has an S-node child.

(P3) For every R-node λ with an R-node child μ, $d(\mu) < \min\{\text{space}(f_1), \text{space}(f_2)\}$ holds, where f_1, f_2 are the two faces incident to the edge of $\text{skel}(\lambda)$ that corresponds to μ.

3.3 Correctness

We now prove the correctness of the construction. Let Γ be an arbitrary embedding of G that is level-planar with demand d. Since $\mathcal{S}$ is the SPQR-tree of G, it represents Γ. By Lemma 6 Γ is also represented by $\mathcal{S}'$ and therefore also by $\mathcal{S}''$, since S-nodes do not provide any embedding options. Finally, Lemma 8, guarantees that all contracted arcs connect two R-nodes that are relatively fixed, and therefore Γ is also represented by $\mathcal{T}$. Conversely, we prove that every planar embedding represented by $\mathcal{T}$ is level-planar with demand d.

Recall that the LP-tree $\mathcal{T}$ can be constructed only if there exists at least one planar embedding Γ_{ref} of G that is level-planar with demand d. By the above arguments, the LP-tree represents all embeddings that are level-planar with demand d, in particular the embedding Γ_{ref}.

Let now Γ be an arbitrary planar embedding of G that is represented by $\mathcal{T}$. Then G induces an embedding $\Gamma(\mu)$ of $G(\mu)$ for each node of μ. The following two lemmas will be useful to prove the correctness. The first one states that each vertex of an R-node skeleton is incident to at least one face whose space value is greater than its demand.

Lemma 9. *Let $\mathcal{T}$ be the LP-tree of G. Let further μ be an R-node of $\mathcal{T}$ and let x be a vertex of* $\text{skel}(\mu)$*. Then there is a face f of* $\text{skel}(\mu)$ *that is incident to x such that $d(x) < \text{space}(f)$.*

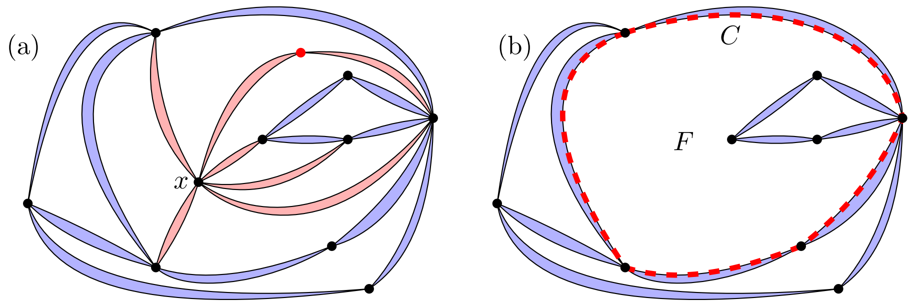

Figure 8: Illustration of the proof of Lemma 9.

Proof. Note that the statement clearly holds if x is incident to the outer face of $\mathrm{skel}(\mu)$, as its space value is ∞ by definition. Hence assume that x is not incident to the outer face of $\mathrm{skel}(\mu)$. Consider an arbitrary embedding Γ that is level-planar with demand d, and which is hence represented by $\mathcal{T}$. If there is a face g of Γ incident to x with $\ell(g) > d(x)$ that projects to a face f of $\mathrm{skel}(\mu)$, then by Lemma 7 $d(x) < \ell(g) = \mathrm{space}(f)$ holds.

Assume for the sake of contradiction that $d(x) \geq \ell(g)$ for each face g of Γ that projects to a face of $\mathrm{skel}(\mu)$ incident to x. Remove from G all vertices (together with their incident edges) that are either (i) a non-pole vertex of some child $G(\nu)$ whose corresponding edge ε of $\mathrm{skel}(\mu)$ is incident only to faces that are also incident to x or (ii) a vertex of $\mathrm{skel}(\mu)$ whose incident faces in $\mathrm{skel}(\mu)$ are all incident to x; see Fig. 8a for an illustration, where the parts that are removed are shaded red. Let G' be the resulting graph and let Γ' be the embedding of G' induced by Γ. Observe that Γ' has an interior face F that used to contain all the removed vertices; see Fig. 8b. Let C be a simple cycle that is entirely contained in the boundary of F and that encloses F in its interior. Observe that, by construction of Γ', each vertex v of C is incident to a face g of Γ that projects to a face of $\mathrm{skel}(\mu)$ incident to x. Therefore $\ell(C) \leq d(x)$. However, in Γ the vertex x lies in the interior of C. Therefore Γ is not level-planar with demand d by Lemma 5. A contradiction. □

The next lemma finally establishes that the space values assigned to the faces of R-node skeletons have a meaning for arbitary planar embeddings represented by the LP-tree, namely the apex of each face g projecting to an inner face f of an R-node skeleton is at least as high as the space value of f promises.

Lemma 10. *Let Γ be a planar embedding of G that is represented by the LP-tree of G. Let μ be an R-node and let f be an inner face of* $\mathrm{skel}(\mu)$*. Then the face g of Γ that projects to f satisfies $\ell(g) \geq \mathrm{space}(f)$.*

Proof. Suppose for the sake of contradiction that $\ell(g) < \mathrm{space}(f)$. Consider a vertex a that is an apex of the face that projects to f in the embedding Γ_{ref}, which is level-planar with demand d. Since $\ell(g) < \mathrm{space}(f) = \ell(a)$, a is not incident to g in Γ. It is therefore not a vertex of $\mathrm{skel}(\mu)$ but belongs to $G(\nu)$ for a unique child ν of μ. Observe that ν cannot be a Q-node as they do not have interior vertices, and it cannot be an S-node by property (P2) since S-node children have been contracted into their R-node parents during the construction of the LP-tree. If ν is a P-node, then its higher pole v satisfies $\ell(a) \leq d(a) < \ell(v)$ by property (P1). But then $\ell(v) > \text{space}(f)$ is a contradiction. Therefore ν must be an R-node. But then $d(\nu) \geq d(a) \geq \ell(a) = \text{space}(f)$, contradicting property (P3). $\square$

We are now ready to prove that an arbitary planar embedding represented by the LP-tree is level-planar with demand d.

Lemma 11. *Let $\mathcal{T}$ be the LP-tree and let Γ be a planar embedding represented by $\mathcal{T}$. Then Γ is level-planar with demand d.*

Proof. If Γ is not level-planar, then by Corollary 1, there exists an interior vertex x such that $d(x) \geq \ell(f)$ for each face f incident to x. Consider the bottom-most node μ of $\mathcal{T}$ for which x is embedded in the interior of $\Gamma(\mu)$. Then either x is an inner vertex of $\text{skel}(\mu)$ or it belongs to $G(\nu)$ for a unique child ν of μ and x is incident to the outer face of $\Gamma(\nu)$. We distinguish cases based on the type of μ.

If μ is a Q-node, then x is one of its poles, and therefore incident to the outer face of μ, contradicting the choice of μ. If μ is an S-node, then x is either a vertex of $\text{skel}(\mu)$, or it lies on the outer face of $\Gamma(\nu)$. In either case, it also lies on the outer face of $\Gamma(\mu)$, contradicting the choice of μ. If μ is a P-node with poles u, v, then x cannot be a vertex of $\text{skel}(\mu)$ and it must lie on the outer face of some $\Gamma(\nu)$. Therefore x and v share a face f. However, we have $d(x) \leq d(\nu) < \ell(v) \leq \ell(f)$ by property (P1). This contradicts the choice of x. It remains to deal with the case that μ is an R-node. We now distinguish cases based on whether x is a vertex of $\text{skel}(\mu)$ or belongs to $G(\nu)$ for some child ν.

If x is a vertex of $\text{skel}(\mu)$, by Lemma 9 there is a face f of $\text{skel}(\mu)$ incident to x with $d(x) < \text{space}(f)$. Consider the face g of Γ that projects to f. Then g is incident to x, and by Lemma 10 $\ell(g) \geq \text{space}(f) > d(x)$ holds. This again contradicts the choice of x. We may hence assume that x belongs to a child $G(\nu)$ and that it lies on the outer face of $\Gamma(\nu)$ but is not one of its poles.

As before, ν cannot be Q-node (for the lack of interior vertices). As above ν cannot be a P-node as otherwise x shares a face with the higher pole v of ν, and $\ell(v) > d(\nu) \geq d(x)$ by property (P1), contradicting the choice of x. Also ν cannot be an S-node by property (P2). So ν must be an R-node.

Let f_1, f_2 be the two faces of $\text{skel}(\mu)$ that are incident to the virtual edge of $\text{skel}(\mu)$ that corresponds to ν. Let further g_1, g_2 be the two faces of Γ that project to f_1 and f_2, respectively. By Lemma 10 $\ell(g_1) \geq \text{space}(f_1)$ and $\ell(g_2) \geq \text{space}(f_2)$. Assume further that x is incident to g_1. By assumption, $d(x) \geq \ell(g_1) \geq \text{space}(f_1)$. But then $d(\nu) \geq d(x) > \text{space}(f_1)$ contradicts property (P3). $\square$

This shows that the LP-tree represents all level-planar embeddings of the graph G. To obtain the following theorem, which is our main result, it remains to prove that the LP-tree can be constructed in linear time.

Theorem 1. *Let G be a biconnected, single-source, level-planar graph. The LP-tree of G represents exactly the level-planar embeddings of G (with demand d) and can be computed in linear time.*

3.4 Construction in Linear Time

Clearly, the construction of the LP-tree described in Section 3.2 can be carried out in polynomial time. In this section, we describe an implementation of it that has linear running time. Starting out, the preprocessing step where the apex t and the edge (s,t) is added to G is feasible in linear time. Next, the SPQR-tree $\mathcal{S}$ of this modified graph G can be computed in linear time [28, 33]. Then, a level-planar embedding Γ_{ref} (with demand d) of G is computed in linear time [19] and all skeletons of $\mathcal{T}$ are embedded accordingly. Note that the demands can be modeled by adding for each vertex v with $\ell(v) < d(v)$ an additional vertex v' with $\ell(v') = d(v)$ and connecting it to v.

For each node μ of $\mathcal{S}$ we need the demand $d(\mu)$. The demands for all nodes are computed bottom-up. For a Q-node, we can set $d(\mu) = -\infty$ (recall that the poles are excluded in the definition. In general, to determine the demand for a node μ of $\mathcal{S}$, proceed as follows. Let C_μ be the set of children of μ and let X be the vertices of $\text{skel}(\mu)$ except for the poles. Then $d(\mu) = \max(\{d(\nu) \mid \nu \in C_\mu\} \cup \{d(x) \mid x \in X\})$. Thus, the running time spent to determine the demand of μ when the demands of all its children are known is linear in the size of $\text{skel}(\mu)$. Since the sum of the sizes of all skeletons of $\mathcal{S}$ is linear in n, all demands can be computed in linear time.

The next step is to split P-nodes. Let μ be a P-node. One split at μ requires to find the child with the greatest demand. Since Γ_{ref} is a level-planar embedding, Lemma 6 gives that this is one of the outermost children. By inspecting the two outermost children of μ, the child ν with greatest demand can be found, or it is found that all children ν of μ satisfy $d(\nu) \leq \ell(v)$ and μ does not need to be split. A P-node split is a constant-time operation. Because there are no more P-node splits than nodes in $\mathcal{S}$, all P-node splits are feasible in linear time. We can therefore compute the decomposition tree $\mathcal{S}'$ in linear time.

The final step of the algorithm are the contractions. First, note that we can contract all S-nodes whose parent is an R-node into their parent skeletons in total linear time to obtain $\mathcal{S}''$. Then, we need to contract edges between R-nodes that are relatively fixed. For this the spaces of all faces of R-nodes need to be known. These can again be computed in a bottom-up manner. Start by labeling every face f of Γ with its apex by walking around the cycle that bounds f. For every edge e of G the apices on both sides of e can then be looked up in Γ. So the incident apices are known for each Q-node of $\mathcal{S}''$. Let μ be a node of $\mathcal{S}''$ so that for each child ν of μ the apices of the ν-incident faces are known. Then the apices of the μ-incident faces can be determined from the child virtual edges of $\text{skel}(\mu)$ that share a face with the parent virtual edge of μ. The running time of this procedure is linear in the sum of sizes of all skeletons, i.e., linear in the size of G. Finally, to contract the arcs, simply traverse S'' top down. For each R-node compute the space around each R-node child node ν from the available apices of the ν-incident faces, compare it with the previously computed demand of $G(\mu)$, and contract the arc (μ, ν) whenever μ and ν turn out to be relatively fixed. Note that the check takes $O(1)$ time per child, and a contraction can be performed in time proportional to the size of the child skeleton. Therefore all contractions can be performed in linear time. This proves the running time claimed in Theorem 1.

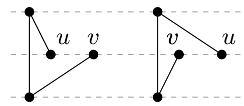

Figure 9: Two level-planar drawings with the same planar embedding but different level orders.

4 Applications

We use the LP-tree to translate efficient algorithms for constrained planarity problems to the level-planar setting. We begin by describing a combinatorial representation of level-planar drawings and their relations to level-planar embedding. Afterwards, we first extend the partial planarity algorithm by Angelini et al. [4] to solve partial level planarity for biconnected single-source level graphs. Second, we adapt this algorithm to solve constrained level planarity. In both cases we obtain a linear-time algorithm, improving upon the best previously known running time of $O(n^2)$, though that algorithm also works in the non-biconnected case [13]. Third, we translate the simultaneous planarity algorithm due to Angelini et al. [5] to the simultaneous level planarity problem when the shared graph is a biconnected single-source level graph. Previously, no polynomial-time algorithm was known for this problem.

4.1 Combinatorial Description of Level-Planar Drawings

For planar graphs, it is customary to work with planar embeddings, i.e., equivalence classes of planar drawings, which can be conveniently represented by the cyclic orders of the edges around vertices plus an outer face.

By contrast, level-planar drawings are often described by giving for each level i the order $\prec_i$ in which the vertices of level i and the edges that cross level i appear along that level from left to right. A collection $\prec = \{\prec_1, \dots, \prec_k\}$ containing an order $\prec_i$ for each level $i = 1, \dots, k$ is a *level order*. A level order is *planar* if there exists a level-planar drawing with these orders. Sometimes it is convenient to subdivide the graph into a *proper* level graph, where $\ell(v) = \ell(u) + 1$ for each edge $(u, v) \in E$. Then the $\prec_i$ are orders of vertices.

It is easy to see that two level-planar drawings Γ_1, Γ_2 can be continuously transformed into each other while keeping vertices at their levels, keeping edges y-monotone, and without introducing intermediate crossings if and only if they have the same level order. Therefore level orders of level-planar graphs are the natural counterpart to embeddings for planar graphs.

It is readily seen that a planar level order $\prec$ uniquely determines a level-planar embedding. The converse is, however, generally not true; see Fig. 9. For single-source level graphs, however, the converse holds and planar level orders are equivalent to level-planar embeddings.

Lemma 12. *For a single-source level graph $G = (V, E)$ the planar level orders correspond* *bijectively to the level-planar embeddings with s on the outer face.*

Proof. Let $G = (V, E)$ be a single-source k-level graph. Assume without loss of generality that $G = (V, E)$ is proper, i.e., $\ell(v) = \ell(u) + 1$ for each edge $(u, v) \in E$. Consider a level-planar drawing Δ of G, let $\prec$ be its level order and let Γ be its embedding.

Let $u, v \in V_i$ be two vertices on level i with $1 \leq i \leq k$. Since G is a single-source graph, there exists be a vertex w of G so that there are disjoint directed paths p_u and p_v from w to u and v, respectively. Let e and f denote the first edge on p_u and p_v, respectively. If w is not the single source of G, it has an incoming edge g. Then $u \prec_i v$ if and only if e, f and g appear in that clockwise order around w in Γ. Otherwise, if $w = s$ is the source of G, let o denote the outer face, which is incident to s. Then, $u \prec_i v$ if and only if o, e and f appear in that clockwise order around w in Γ. The claim then follows easily. □

4.2 Partial Level Planarity

Angelini et al. [4] study the problem of extending a partial drawing in the planar setting. Given a graph G along with a planar drawing Δ_H of a subgraph $H \subseteq G$, the question is whether there exists a planar drawing Δ_G of G whose restriction to H coincides with Δ_H. Angelini et al. show that, for planar drawings, this can be rephrased as a combinatorial embedding problem. They hence define a *partially embedded graph* as a triple (G, H, Γ_H) that consists of a graph G, a subgraph $H \subseteq G$, and a planar embedding Γ_H of H. A partially embedded graph is *planar* if and only if there exists a planar embedding Γ_G of G whose restriction to H coincides with Γ_H. In this case Γ_G is also called an *extension* of (G, H, Γ_H).

We consider an analogous partial drawing extension problem for level-planar graphs. A *partially level-ordered graph* is a triple $(G, H, \prec_H)$ consisting of a graph G, a subgraph $H \subseteq G$, and a planar level order $\prec_H$ of H. As above, $(G, H, \prec_H)$ is *level-planar* if and only if there exists a planar level order $\prec_G$ of G whose restriction to H coincides with H. Let Γ_H be the level-planar embedding of H induced by $\prec_H$. Using Lemma 12, the problem of testing whether a given partially ordered level graph can be rephrased into an embedding problem if H has a single source.

Lemma 13. *If H has a single source, then $(G, H, \prec_H)$ is level-planar if and only if the instance (G, H, Γ_H) is planar and it has an extension that is level-planar.*

Observe that, while the algorithm of Angelini et al. can be used to test whether (G, H, Γ_H) is planar, there seems no easy way to test whether (G, H, Γ_H) has a level-planar extension. It is here that the LP-tree enters the scene. Namely, in case G is biconnected, what Angelini et al. do, is to compute the SPQR-tree $\mathcal{T}$ of G and to search it for a planar embedding that extends Γ_H. Inspecting their algorithm, it can be seen that it only relies on the facts that the skeletons of P-nodes have parallel edges that can be permuted arbitrarily and that the skeletons of R-nodes are biconnected and come with a fixed embedding that may only be flipped. Therefore, by simply using the LP-tree $\mathcal{T}'$ as a drop-in replacement for the SPQR-tree $\mathcal{T}$ in this algorithm without further modifications, it will search for an embedding extension among all level-planar embeddings.

Observe that the use of the LP-tree entails further conditions on G. Besides being biconnected, it is required that G has a single source s and a unique apex t along with the edge (s, t). We note that only the requirement of a single source is a restriction, whereas the remaining two conditions can be established as described in Section 2. Altogether we have the following theorem.

Theorem 2. *Let $(G, H, \prec_H)$ be a partially level-ordered graph such that G and H are single-source graphs and G is biconnected. It can be tested in linear time whether $(G, H, \prec_H)$ is level-planar.*

Observe that, to make this work, it is necessary to compute Γ_H from $\prec_H$ in linear time. We prove that this is possible in a more general context in Lemma 15 in the next section.

In what follows, we briefly discuss the restrictions imposed by the above theorem. First note that G being a single-source graph is an absolute prerequisite as the drawing extension problem for level-planar graphs is NP-complete without this restriction [13].

Concerning the biconnectivity of G, we note that Angelini et al. [4] extend their algorithm to the case where G is connected but not necessarily biconnected. This requires significant additional effort and the use of another data structure, called the enriched block-cut tree, that manages the biconnected components of a graph in a tree. Some of the techniques described in this paper, in particular our notion of demands, may be helpful in extending our algorithm to the connected single-source case. Consider a connected single-source graph G. All biconnected components of G have a single source and the LP-tree can be used to represent their level-planar embeddings. However, a vertex v of some biconnected component H of G may be a cutvertex in G and can dominate vertices that do not belong to H. Depending on the space around v and the levels on which these vertices lie, this may restrict the admissible level-planar embeddings of H. Let $X(v)$ denote the set of vertices dominated by v that do not belong to H. Set the demand of v to $d(v) = d(X(v))$. Computing the LP-tree with these demands ensures that there is enough space around each cutvertex v to embed all components connected at v. The remaining choices are into which faces of H incident to v such components can be embedded and possibly nesting biconnected components. These choices are largely independent for different components and only depend on the available space in each incident face. This information is known from the LP-tree computation. In this way it may be possible to extend the steps for handling non-biconnected graphs due to Angelini et al. to the level-planar setting.

Finally, the requirement that H has a single source was only necessary to rewrite the drawing extension problem into an embedding extension problem. This can in fact be alleviated as the next section shows, which solves a more general problem.

4.3 Constrained Level Planarity

A *constrained level graph* is a tuple $(G, \prec')$ that consists of a k-level graph $G = (V, E)$ and a set $\prec' = \{\prec'_1, \prec'_2, \dots, \prec'_k\}$ of irreflexive relations $\prec'_i \subseteq V_i \times V_i$ for $i = 1, \dots, k$ [13]. The task is to find a drawing of G, i.e., a planar level order $\prec$ of G such that for any two vertices $u, v \in V_i$ we have that $u \prec'_i v$ implies $u \prec_i v$. If this is the case, then $(G, \prec')$ is called *level-planar*. Brückner and Rutter [13] showed that testing level-planarity of partially level-ordered graphs is a special case of testing level-planarity of constrained level graphs.

If G has a single source s, then similarly to Lemma 12, we can transfer the partial order constraints into constraints on a level-planar embedding of G. To this end, consider two vertices u, v on the same level i such that $u \prec'_i v$. Let w be a vertex so that G contains disjoint directed paths p_u and p_v from w to u and v, respectively. Note that such a vertex w exists since G has a single source s. As described in Section 2, we may further assume that G has a unique apex t and it contains the edge (s, t).

Let e and f denote the first edge of p_u and p_v, respectively. Further, if $w \neq s$, let g be an arbitrary incoming edge of w, otherwise let $g = (s, t)$. We define $T(u \prec'_i v) = (e, f, g)$ and call this a *triple constraint*. Along the lines of the proof of Lemma 12 it is seen that in a level-planar drawing $\prec$ of G we have $u \prec_i v$ if and only if in the corresponding level-planar embedding Γ of G the edges (e, f, g) occur in this clockwise order around w; in this case, we say that Γ *satisfies* the triple constraint (e, f, g).

Let $T(\prec') = \{T(u \prec'_i v) \mid u, v \in V_i \text{ for } i \in \{1, \dots, k\} \text{ and } u \prec'_i v\}$ be the triple constraints for all comparable pairs of V. We say that a level-planar embedding Γ of G *satisfies* $T(\prec')$ if it satisfies all triple constraints in $T(\prec')$. The above discussion implies the following characterization.

Lemma 14. *Let $(G, \prec')$ be a level graph with a single source s, a unique apex t, and that contains the edge (s, t). Then $(G, \prec')$ is level-planar if and only if G admits a level-planar embedding Γ that satisfies $T(\prec')$.*

The following lemma shows that $T(\prec')$ can be computed in linear time.

Lemma 15. *Given $\prec'$, $T(\prec')$ can be computed in linear time.*

Proof. We start by finding for each pair u, v with $u \prec'_i v$ a vertex w so that there are disjoint paths p_u and p_v from w to u and v. This can be achieved in linear time by using the algorithm of Harel and Tarjan on a depth-first-search tree $\mathcal{D}$ of G [31]. Mark w with the pair (u, v) for the next step. Then, we find the edges e and f of p_u and p_v incident to w, respectively. To this end, we proceed similarly to a technique described by Bläsius et al. [9]. At the beginning, every vertex of G belongs to its own singleton set. Proceed to process the vertices of G bottom-up in $\mathcal{D}$, i.e., starting from the vertices on the greatest level. When encountering a vertex w marked with a pair (u, v), find the representatives of u and v, denoted by u' and v', respectively. Observe that $e = (w, u')$ and $f = (w, v')$, and that both e and f are tree edges of $\mathcal{D}$. Then unify the sets of all of its direct descendants in $\mathcal{D}$ and let w be the representative of the resulting union. Because all union operations are known in advance we can use the linear-time union-find algorithm of Gabow and Tarjan [26]. Finally, pick as g the unique incoming edge of w in $\mathcal{D}$, or the edge (s, t) if $w = s$. Altogether, we obtain $T(\prec')$ in time linear in G and the number of comparable pairs $u \prec'_i v$ we consider. □

Theorem 3. *Constrained level planarity can be solved in linear running time for biconnected single-source level graphs.*

Proof. Let $(G, \prec')$ be a constrained level graph such that G is biconnected and has a single source. As in Section 2, we may assume that G has a unique apex t and it contains the edge (s, t). We compute $T(\prec')$ in linear time by Lemma 15. According to Lemma 14 it suffices to check whether G admits a level-planar embedding that satisfies $T(\prec')$.

Consider a triple (e, f, g) of edges sharing a common vertex w and let $\mathcal{T}$ be the LP-tree of G. Then e, f, g correspond to distinct leaves of $\mathcal{T}$, and there is a unique node μ for which e, f, g are contained in distinct connected components of of $T - \mu$. It follows that in skel(μ) the edges e, f, g are represented by distinct virtual edges, and the embedding of skel(μ) determines the circular order of (e, f, g) around w. Therefore, each triple constraint (e, f, g) corresponds to a triple constraint (e_μ, f_μ, g_μ) in some skeleton μ of $\mathcal{T}$. We say that (e, f, g) projects to μ and that (e_μ, f_μ, g_μ) is its projection to skel(μ). It hence follows that, to satisfy $T(\prec')$, it suffices to find for each node μ of the LP-tree an embedding that satisfies the projections of all triple constraints in $T(\prec')$ that project to μ. We note that Angelini et al. [4] use the same observation on SPQR-trees for their embedding extension algorithm. Using techniques similar to the ones by Bläsius et al. [9] or Da Lozzo and Rutter [18] for SPQR-trees, the projections of all triple constraints can be computed in time linear in the sizes of G and $T(\prec')$. It then remains to determine suitable embeddings for the skeletons of $\mathcal{T}$.

To this end, recall that only R- and P-nodes offer an embedding choice and that for each R-node μ, the skeleton skel(μ) has a unique embedding up to a flip. For an R-node μ, it can be easily tested in time linear in the size of skel(μ) and the number of constraints that project to μ whether one of these two flips satisfies all the triples that project to μ. For a P-node μ let u, v denote its two poles with $\ell(u) < \ell(v)$. Observe that by property (P1) only the parent edge may represent outgoing edges of v. Since each triple constraint contains two outgoing and one incoming edge of the shared vertex, it follows that all constraints that project to skel(μ) affect u. Note that, generally, finding a circular order around v satisfying given triples is equivalent to the NP-complete problem CyclicOrdering [27]. However, the property that G has a single source guarantees that only the parent edge may represent edges incoming at u, and therefore each triple constraint $(e, f, g) \in T(\prec')$ that projects to (e_μ, f_μ, g_μ) in μ contains the parent edge as g_μ. By convention, the parent edge is embedded left-most, and it therefore suffices to find an embedding where e_μ is embedded before f_μ for each such constraint. This corresponds to a simple topological ordering problem that can be solved in time linear in the size of skel(μ) and the number of triple constraints that project to μ. □

4.4 Simultaneous Level Planarity

We translate the simultaneous planarity algorithm of Angelini et al. [5] to solve simultaneous level planarity for biconnected single-source graphs. They define simultaneous planarity as follows. Let $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ be two graphs with the same vertices. The *shared edges* $E_1 \cap E_2$ together with V make up the intersection graph $G_{1\cap 2}$, or simply G for short. All other edges are *exclusive*. The graphs G_1 and G_2 are *simultaneously planar* if there exist planar drawings Γ_1 and Γ_2 of G_1 and G_2, respectively, whose restrictions to the shared graph G coincide. Jünger and Schulz [37] showed that the problem can be

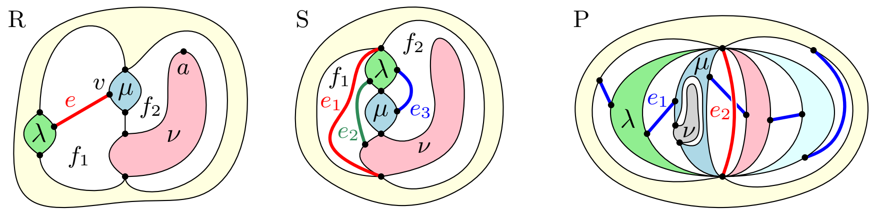

Figure 10: In the R-node, e fixes the relative embeddings of $G(\lambda)$ and $G(\mu)$. In the level-planar setting, e also fixes the embedding of $G(\nu)$. In the S-node, e_2 and e_3 fix the relative embeddings of $G(\lambda), G(\nu)$ and $G(\lambda), G(\mu)$, respectively. In the level-planar setting, e_1 also fixes the embedding of $G(\nu)$. In the P-node, e_1 fixes the relative embeddings of $G(\lambda)$ and $G(\mu)$. In the level-planar setting, e_1 also fixes the embedding of $G(\nu)$.

equivalently phrased in terms of embeddings. Namely, two graphs G_1, G_2 on the same vertex set are simultaneously planar if and only if there exist planar embeddings of G_1 and G_2 that induce the same embedding on G. Or, equivalently, the question is whether there exists a planar embedding of G that simultaneously extends to planar embeddings of G_1 and of G_2. Angelini et al. [5] give a linear-time algorithm for this problem under the condition that the shared graph G is biconnected.

We study the analogous problem for level planarity. Given two graphs G_1, G_2 with shared graph G as above, are there level-planar drawings of G_1, G_2 that coincide on the shared graph? Naturally, this is equivalent to asking whether G_1, G_2 admit planar level orders that coincide on the shared graph. Assuming that G (and therefore also G_1, G_2) have a single source s and using Lemma 12, this is equivalent to asking whether there exist level-planar embeddings of G_1 and G_2 that coincide on the shared graph. As above, we can rephrase this as asking whether the shared graph G admits a level-planar embedding that simultaneously extends to level-planar embeddings of G_1 and of G_2.

Considering the algorithm of Angelini et al. [5] one observes that it finds a suitable embedding of G by exploiting the SPQR-tree $\mathcal{T}$ of G, and that it relies only on the usual properties that children of P-nodes may be permuted arbitrarily and skeletons of R-nodes are biconnected and come with a fixed embedding up to a flip. Thus, running the very same algorithm but on the LP-tree $\mathcal{T}'$ of G, would determine the existence of a level-planar embedding Γ of G that simultaneously extends to planar embeddings of G_1 and of G_2. Unfortunately, this is not quite enough, since answering the simultaneous level-planarity question requires that Γ simultaneously extends to *level-planar* embeddings of G_1 and G_2.

In the following we sketch how to adapt the algorithm of Angelini et al. [5] to achieve this. As mentioned above, their algorithm works by constructing the SPQR-tree for the shared graph G. It then expresses the constraints imposed on G by the exclusive edges as a 2-Sat instance S that is satisfiable if and only if G_1 and G_2 admit a simultaneous embedding. We give a very brief overview of the 2-Sat constraints in the planar setting. In an R-node, an exclusive edge e has to be embedded into a unique face. This potentially restricts the embedding of the expansion graphs $G(\lambda), G(\mu)$ that contain the endpoints of e, i.e., the embedding of $G(\lambda)$ and $G(\mu)$ is fixed with respect to the embedding of the R-node. Add a variable x_μ to S for every node of $\mathcal{T}$ with the semantics that x_μ is true if skel(μ)

has its reference embedding Γ_μ, and false if the embedding of skel(μ) is the reflection of Γ_μ. The restriction imposed by e on $G(\lambda)$ and $G(\mu)$ can then be modeled as a 2-Sat constraint on the variables x_λ and x_μ. For example, in the R-node shown in Fig. 10 on the left, the internal edge e must be embedded into face f_1, which fixes the relative embeddings of $G(\lambda)$ and $G(\mu)$. In an S-node, an exclusive edge e may be embedded into one of the two candidate faces f_1, f_2 around the node. The edge e can conflict with another exclusive edge e' of the S-node, meaning that e and e' cannot be embedded in the same face. This is modeled by introducing for every exclusive edge e and candidate face f the variable x_e^f with the semantics that x_e^f is true iff e is embedded into f. The previously mentioned conflict can then be resolved by adding the constraints $x_e^{f_1} \vee x_e^{f_2}$, $x_{e'}^{f_1} \vee x_{e'}^{f_2}$ and $x_e^{f_1} \neq x_{e'}^{f_1}$ to S. Additionally, an exclusive edge e whose endpoints lie in different expansion graphs can restrict their respective embeddings. For example, in the S-node shown in Fig. 10 in the middle, the edges e_2 and e_3 may not be embedded into the same face. And e_2 and e_3 fix the embeddings of $G(\lambda)$ and $G(\nu)$ and of $G(\lambda)$ and $G(\mu)$, respectively. This would be modeled as $x_\lambda = x_\nu$ and $x_\lambda = x_\mu$ in S. In a P-node, an exclusive edge can restrict the embeddings of expansion graphs just like in R-nodes. Additionally, exclusive edges between the poles of a P-node can always be embedded unless all virtual edges are forced to be adjacent by internal edges. For example, in the P-node shown in Fig. 10 on the right, e_1 fixes the relative embeddings of $G(\lambda)$ and $G(\mu)$. And e_2 can be embedded iff one of the blue edges does not exist.

We now adapt the algorithm to the level-planar setting. First, replace the SPQR-tree with the LP-tree $\mathcal{T}$. The satisfying truth assignments of S then correspond to simultaneous planar embeddings $\mathcal{E}_1, \mathcal{E}_2$ of G_1, G_2, so that their shared embedding $\mathcal{E}$ of G is level planar. However, due to the presence of exclusive edges, $\mathcal{E}_1$ and $\mathcal{E}_2$ are not necessarily level planar. To ensure that $\mathcal{E}_1$ and $\mathcal{E}_2$ are level planar, we add more constraints to S. Consider adding an exclusive edge e into a face f. This splits f into two faces f', f''. The apex of at least one face, say f'', remains unchanged. As a consequence, the space around any virtual edge incident to f'' remains unchanged as well. But the apex of f' can change, namely, the apex of f' is an endpoint of e. Then the space around the virtual edges incident to f' can decrease. This may lower the level of the apices in the faces around the virtual edge associated with ν, which, in the same way as described in Section 3.1, may make additional pairs of adjacent R-nodes relatively fixed, thus limiting their ability to flip independently. This can be described as an implication on the variables x_e^f and x_ν. For an example, see Fig. 10. In the R-node, adding the edge e with endpoint v into f_1 creates a new face f_1' with apex v. This forces $G(\nu)$ to be embedded so that its apex a is embedded into face f_2. Similarly, in the S-node and in the P-node, adding the edge e_1 restricts $G(\nu)$. We collect all these additional implications of embedding e into f and add them to the 2-Sat instance S. Each exclusive edge leads to a constant number of 2-Sat implications. To find each such implication $O(n)$ time is needed in the worst case. Because there are at most $O(n)$ exclusive edges this gives quadratic running time overall. Clearly, all implications must be satisfied for $\mathcal{E}_1$ and $\mathcal{E}_2$ to be level planar. On the other hand, suppose that one of $\mathcal{E}_1$ or $\mathcal{E}_2$, say $\mathcal{E}_1$, is not level planar. Because the restriction of $\mathcal{E}_1$ to G is level-planar due to the LP-tree and planar due to the algorithm by Angelini et al., there must be a crossing involving an exclusive edge e of G_1. This contradicts the fact that we have respected all necessary implications of embedding e. We obtain Theorem 4.

Theorem 4. *Simultaneous level planarity can be solved in quadratic time for two graphs whose intersection is a biconnected single-source level graph.*

In the non-biconnected setting Angelini et al. solve the case when the intersection graph is a star. Haeupler et al. describe an algorithm for simultaneous planarity that does not use SPQR-trees, but they also require biconnectivity [30]. Very recently, the case where G is connected has been solved with a running time of $O(n^8)$ [25], which was subsequently improved to $O(n^2)$ [8]. However the techniques are quite different and it is unclear whether LP-trees can be leveraged to obtain similar results for level-planar graphs.

5 Conclusion

The majority of constrained embedding algorithms for planar graphs rely on two features of the SPQR-tree: they are decomposition trees and the embedding choices consist of arbitrarily permuting parallel edges between two poles and choosing the flip of a skeleton whose embedding is unique up to reflection. We have developed the LP-tree, an SPQR-tree-like embedding representation that has both of these features. An SPQR-tree-based algorithm that tests whether a biconnected graph G has a planar embedding satisfying a certain property $\mathcal{P}$ can then usually be executed on LP-trees without any modification to determine whether a given biconnected single-source graph G has a level-planar embedding satisfying property $\mathcal{P}$. The necessity for mostly minor modifications only stems from the fact that in many cases the level-planar version of a drawing problem imposes additional restrictions on the embedding compared to the original planar version, i.e., one seeks a level-planar embedding of G that satisfies certain properties $\mathcal{P}'$ that are usually a superset of $\mathcal{P}$. Our LP-tree thus allows to leverage a large body of literature on constrained embedding problems and to transfer it to the level-planar setting.

In particular, we have used it to obtain linear-time algorithms for partial and constrained level planarity in the biconnected single-source case, which improves upon the previous best known running time of $O(n^2)$. Moreover, we have presented an efficient algorithm for the simultaneous level planarity problem when the shared graph is biconnected and has a single source. Previously, no polynomial-time algorithm was known for this problem.

It is an interesting question whether our results can be extended to level-planar graphs with multiple sources. However, a data structure with properties similar to the LP-tree that represents level-planar embeddings of graphs with multiple sources could be used to efficiently solve the partial drawing extension problem for level-planar graphs with multiple sources, which is known to be NP-complete [13]. It hence follows that, most likely, such a data structure either does not exist or cannot be constructed efficiently. It therefore seems a more promising question to see whether our LP-tree based techniques can be used to obtain FPT algorithms with respect to the number of sources.

Acknowledgments. We thank the anonymous reviewers of the journal version of this article for comments that inspired a significant simplifcation of the results and that helped us to improve the presentation.

References

[1] Patrizio Angelini and Michael A. Bekos. Hierarchical partial planarity. *Algorithmica*, 81(6):2196–2221, 2019. doi:10.1007/s00453-018-0530-6.

[2] Patrizio Angelini, Thomas Bläsius, and Ignaz Rutter. Testing mutual duality of planar graphs. *Int. J. Comput. Geom. Appl.*, 24(4):325–346, 2014. doi:10.1142/S0218195914600103.

[3] Patrizio Angelini, Steven Chaplick, Sabine Cornelsen, Giordano Da Lozzo, Giuseppe Di Battista, Peter Eades, Philipp Kindermann, Jan Kratochvíl, Fabian Lipp, and Ignaz Rutter. Simultaneous orthogonal planarity. In Yifan Hu and Martin Nöllenburg, editors, *Proceedings of the 24th International Symposium on Graph Drawing and Network Visualization (GD'16)*, volume 9801 of *Lecture Notes in Computer Science*, pages 532–545. Springer, 2016. doi:10.1007/978-3-319-50106-2_41.

[4] Patrizio Angelini, Giuseppe Di Battista, Fabrizio Frati, Vít Jelínek, Jan Kratochvíl, Maurizio Patrignani, and Ignaz Rutter. Testing planarity of partially embedded graphs. *ACM Trans. Algorithms*, 11(4):32:1–32:42, 2015. doi:10.1145/2629341.

[5] Patrizio Angelini, Giuseppe Di Battista, Fabrizio Frati, Maurizio Patrignani, and Ignaz Rutter. Testing the simultaneous embeddability of two graphs whose intersection is a biconnected or a connected graph. *J. Discrete Algorithms*, 14:150–172, 2012. doi:10.1016/j.jda.2011.12.015.

[6] Patrizio Angelini, Giuseppe Di Battista, and Maurizio Patrignani. Finding a minimum-depth embedding of a planar graph in $O(n^4)$ time. *Algorithmica*, 60(4):890–937, 2011. doi:10.1007/s00453-009-9380-6.

[7] Daniel Bienstock and Clyde L. Monma. On the complexity of embedding planar graphs to minimize certain distance measures. *Algorithmica*, 5(1):93–109, 1990. doi:10.1007/BF01840379.

[8] Thomas Bläsius, Simon D. Fink, and Ignaz Rutter. Synchronized planarity with applications to constrained planarity problems. In Petra Mutzel, Rasmus Pagh, and Grzegorz Herman, editors, *Proceedings of the 29th Annual European Symposium on Algorithms (ESA'21)*, volume 204 of *LIPIcs*, pages 19:1–19:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ESA.2021.19.

[9] Thomas Bläsius, Annette Karrer, and Ignaz Rutter. Simultaneous embedding: Edge orderings, relative positions, cutvertices. *Algorithmica*, 80(4):1214–1277, 2018. doi:10.1007/s00453-017-0301-9.

[10] Thomas Bläsius, Stephen G. Kobourov, and Ignaz Rutter. Simultaneous embedding of planar graphs. In Roberto Tamassia, editor, *Handbook on Graph Drawing and Visualization*, pages 349–381. Chapman and Hall/CRC, 2013.

[11] Thomas Bläsius, Ignaz Rutter, and Dorothea Wagner. Optimal orthogonal graph drawing with convex bend costs. *ACM Trans. Algorithms*, 12(3):33:1–33:32, 2016. doi:10.1145/2838736.

[12] Guido Brückner, Markus Himmel, and Ignaz Rutter. An SPQR-tree-like embedding representation for upward planarity. In Daniel Archambault and Csaba D. Tóth, editors, *Proceedings of the 27th International Symposium on Graph Drawing and Network Visualization (GD'19)*, volume 11904 of *Lecture Notes in Computer Science*, pages 517–531. Springer, 2019. doi:10.1007/978-3-030-35802-0_39.

[13] Guido Brückner and Ignaz Rutter. Partial and constrained level planarity. In Philip N. Klein, editor, *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'17)*, pages 2000–2011. SIAM, 2017. doi:10.1137/1.9781611974782.130.

[14] Norishige Chiba, Takao Nishizeki, Shigenobu Abe, and Takao Ozawa. A linear algorithm for embedding planar graphs using PQ-trees. *J. Comput. Syst. Sci.*, 30(1):54–76, 1985. doi:10.1016/0022-0000(85)90004-2.

[15] Markus Chimani, Carsten Gutwenger, Petra Mutzel, and Christian Wolf. Inserting a vertex into a planar graph. In Claire Mathieu, editor, *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'09)*, pages 375–383. SIAM, 2009. URL: http://dl.acm.org/citation.cfm?id=1496770.1496812.

[16] Markus Chimani and Petr Hlinený. Inserting multiple edges into a planar graph. In Sándor P. Fekete and Anna Lubiw, editors, *Proceedings of the 32nd International Symposium on Computational Geometry (SOCG'16)*, volume 51 of *LIPIcs*, pages 30:1–30:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.SoCG.2016.30.

[17] Giordano Da Lozzo, Giuseppe Di Battista, and Fabrizio Frati. Extending upward planar graph drawings. *Comput. Geom.*, 91:101668, 2020. doi:10.1016/j.comgeo.2020.101668.

[18] Giordano Da Lozzo and Ignaz Rutter. Planarity of streamed graphs. *Theor. Comput. Sci.*, 799:1–21, 2019. doi:10.1016/j.tcs.2019.09.029.

[19] Giuseppe Di Battista and Enrico Nardelli. Hierarchies and planarity theory. *IEEE Trans. Syst. Man Cybern.*, 18(6):1035–1046, 1988. doi:10.1109/21.23105.

[20] Giuseppe Di Battista and Roberto Tamassia. Algorithms for plane representations of acyclic digraphs. *Theor. Comput. Sci.*, 61:175–198, 1988. doi:10.1016/0304-3975(88)90123-5.

[21] Giuseppe Di Battista and Roberto Tamassia. Incremental planarity testing (extended abstract). In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science (FOCS'89)*, pages 436–441. IEEE Computer Society, 1989. doi:10.1109/SFCS.1989.63515.

[22] Giuseppe Di Battista and Roberto Tamassia. On-line graph algorithms with SPQR-trees. In Mike Paterson, editor, *Proceedings of the 17th International Colloquium on Automata, Languages and Programming (ICALP'90)*, volume 443 of *Lecture Notes in Computer Science*, pages 598–611. Springer, 1990. doi:10.1007/BFb0032061.

[23] Giuseppe Di Battista and Roberto Tamassia. On-line maintenance of triconnected components with SPQR-trees. *Algorithmica*, 15(4):302–318, 1996. doi:10.1007/BF01961541.

[24] Radoslav Fulek, Michael J. Pelsmajer, Marcus Schaefer, and Daniel Stefankovic. Hanani-tutte and monotone drawings. In Petr Kolman and Jan Kratochvíl, editors, *Proceedings of the 37th International Workshop on Graph-Theoretic Concepts in Computer Science (WG'11)*, volume 6986 of *Lecture Notes in Computer Science*, pages 283–294. Springer, 2011. doi:10.1007/978-3-642-25870-1_26.

[25] Radoslav Fulek and Csaba D. Tóth. Atomic embeddability, clustered planarity, and thickenability. *J. ACM*, 69(2):13:1–13:34, 2022. doi:10.1145/3502264.

[26] Harold N. Gabow and Robert Endre Tarjan. A linear-time algorithm for a special case of disjoint set union. *J. Comput. Syst. Sci.*, 30(2):209–221, 1985. doi:10.1016/0022-0000(85)90014-5.

[27] Zvi Galil and Nimrod Megiddo. Cyclic ordering is np-complete. *Theor. Comput. Sci.*, 5(2):179–182, 1977. doi:10.1016/0304-3975(77)90005-6.

[28] Carsten Gutwenger and Petra Mutzel. A linear time implementation of SPQR-trees. In Joe Marks, editor, *Proceedings of the 8th International Symposium on Graph Drawing (GD'00)*, volume 1984 of *Lecture Notes in Computer Science*, pages 77–90. Springer, 2000. doi:10.1007/3-540-44541-2_8.

[29] Carsten Gutwenger, Petra Mutzel, and René Weiskircher. Inserting an edge into a planar graph. *Algorithmica*, 41(4):289–308, 2005. doi:10.1007/s00453-004-1128-8.

[30] Bernhard Haeupler, Krishnam Raju Jampani, and Anna Lubiw. Testing simultaneous planarity when the common graph is 2-connected. *J. Graph Algorithms Appl.*, 17(3):147–171, 2013. doi:10.7155/jgaa.00289.

[31] Dov Harel and Robert Endre Tarjan. Fast algorithms for finding nearest common ancestors. *SIAM J. Comput.*, 13(2):338–355, 1984. doi:10.1137/0213024.

[32] Seok-Hee Hong, Brendan D. McKay, and Peter Eades. A linear time algorithm for constructing maximally symmetric straight line drawings of triconnected planar graphs. *Discret. Comput. Geom.*, 36(2):283–311, 2006. doi:10.1007/s00454-006-1231-5.

[33] John E. Hopcroft and Robert Endre Tarjan. Dividing a graph into triconnected components. *SIAM J. Comput.*, 2(3):135–158, 1973. doi:10.1137/0202012.

[34] Michael D. Hutton and Anna Lubiw. Upward planar drawing of single-source acyclic digraphs. *SIAM J. Comput.*, 25(2):291–311, 1996. doi:10.1137/S0097539792235906.

[35] Michael Jünger and Sebastian Leipert. Level planar embedding in linear time. *J. Graph Algorithms Appl.*, 6(1):67–113, 2002. doi:10.7155/jgaa.00045.

[36] Michael Jünger, Sebastian Leipert, and Petra Mutzel. Level planarity testing in linear time. In Sue Whitesides, editor, *Proceedings of the 6th International Symposium on Graph Drawing (GD'98)*, volume 1547 of *Lecture Notes in Computer Science*, pages 224–237. Springer, 1998. doi:10.1007/3-540-37623-2_17.

[37] Michael Jünger and Michael Schulz. Intersection graphs in simultaneous embedding with fixed edges. *J. Graph Algorithms Appl.*, 13(2):205–218, 2009. doi:10.7155/jgaa.00184.

[38] Saunders Mac Lane. A structural characterization of planar combinatorial graphs. *Duke Mathematical Journal*, 3(3):460–472, 1937. doi:10.1215/S0012-7094-37-00336-3.

[39] Bert Randerath, Ewald Speckenmeyer, Endre Boros, Peter L. Hammer, Alexander Kogan, Kazuhisa Makino, Bruno Simeone, and Ondrej Cepek. A satisfiability formulation of problems on level graphs. *Electron. Notes Discret. Math.*, 9:269–277, 2001. doi:10.1016/S1571-0653(04)00327-0.

[40] Ignaz Rutter. Simultaneous embedding. In Seok-Hee Hong and Takeshi Tokuyama, editors, *Beyond Planar Graphs, Communications of NII Shonan Meetings*, pages 237–265. Springer, 2020. doi:10.1007/978-981-15-6533-5_13.

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

[42] William Thomas Tutte. *Connectivity in Graphs*. University of Toronto Press, 1966.

[43] Hassler Whitney. 2-isomorphic graphs. *American Journal of Mathematics*, 55(1):245–254, 1933.