Clustered Planarity Variants for Level Graphs*

Simon D. Fink[1,2], Matthias Pfretzschner[2], Ignaz Rutter[2], and Marie Diana Sieper[3]

1 **Algorithms and Complexity Group, Technische Universität Wien, Austria**
sfink@ac.tuwien.ac.at

2 **Faculty of Computer Science and Mathematics, University of Passau, Germany**
{finksim,pfretzschner,rutter}@fim.uni-passau.de

3 **Institute of Computer Science, University of Würzburg, Germany**
marie.sieper@uni-wuerzburg.de

Abstract

We consider variants of the clustered planarity problem for level-planar drawings. So far, only convex clusters have been studied in this setting. We introduce two new variants that both insist on a level-planar drawing of the input graph but relax the requirements on the shape of the clusters. In unrestricted Clustered Level Planarity (uCLP) we only require that they are bounded by simple closed curves that enclose exactly the vertices of the cluster and cross each edge of the graph at most once. The problem y-monotone Clustered Level Planarity (y-CLP) requires that additionally it must be possible to augment each cluster with edges that do not cross the cluster boundaries so that it becomes connected while the graph remains level-planar, thereby mimicking a classic characterization of clustered planarity in the level-planar setting.

We give a polynomial-time algorithm for uCLP if the input graph is biconnected and has a single source. By contrast, we show that y-CLP is hard under the same restrictions and it remains NP-hard even if the number of levels is bounded by a constant and there is only a single non-trivial cluster.

Related Version *full version including missing proofs*: arXiv:2402.13153

1 Introduction

A *level graph* (G, γ) is a graph $G = (V, E)$ and a function $\gamma \colon V \to \{1, 2, \dots, k\}$ with $k \in \mathbb{N}$ that assigns vertices to levels such that no two adjacent vertices are assigned to the same level. A *level planar drawing* of a level graph (G, γ) is a crossing-free drawing of G that maps each vertex v to a point on the line $y = \gamma(v)$ and each edge to a y-monotone curve between its endpoints. A level graph is *level planar* if it has a level planar drawing. Level planarity can be tested in linear time [11].

Let $G = (V, E)$ be a graph. A *clustering* T of G is a rooted tree whose leaves are the vertices V. Each inner node μ of T represents a *cluster*, which encompasses all leaves V_μ of the subtree rooted at μ. The pair (G, T) is called a *clustered graph*. A *clustered planar drawing* of a clustered graph (G, T) is a planar drawing of G that also maps every cluster μ to a region R_μ that is enclosed by a simple closed curve such that (i) R_μ contains exactly the vertices V_μ, (ii) no two region boundaries intersect, and (iii) no edge intersects the boundary of a cluster region more than once. The combination of (i) and (iii) implies that an edge may intersect a cluster boundary if and only if precisely one of its endpoints lies inside the cluster. A clustered graph is *clustered planar* if it has a clustered planar drawing. The problem of testing this property and finding such drawings is called Clustered Planarity. In a

* Simon D. Fink was funded by the Deutsche Forschungsgemeinschaft (German Research Foundation, DFG) grant RU-1903/3-1 and by the Vienna Science and Technology Fund (WWTF) [10.47379/ICT22029].

40th European Workshop on Computational Geometry, Ioannina, Greece, March 13–15, 2024.

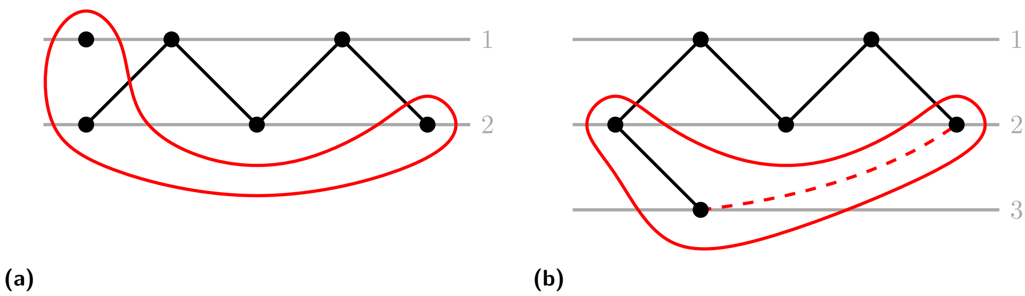

Figure 1 **(a)** A drawing that is level planar and clustered planar and thus cl-planar, but not convex cl-planar or y-cl-planar. **(b)** A drawing that is y-cl-planar (with the augmentation edge in E' shown dashed in red) and thus also cl-planar, but not convex cl-planar.

recent breakthrough, Fulek and Tóth gave the first efficient algorithm for this problem [9], which was soon after improved to a quadratic-time solution [2].

In this paper, we seek to explore the combination of the two concepts of level planarity and clustered planarity. Namely, our input is a *clustered level graph (cl-graph)*, which is a tuple (G, γ, T) such that (G, γ) is a level graph and (G, T) is a clustered graph. We insist on a level-planar drawing of G. However, it is not immediately clear which conditions the cluster boundaries should fulfill. Forster and Bachmaier [8] proposed the problem variant Convex Clustered Level Planarity (short cCLP), which requires to draw the clusters as convex regions[1]. They showed that cCLP can be solved in linear time if the graph is proper (i.e., all edges connect vertices on adjacent levels) and the clusters are level-connected (i.e., each cluster contains an edge between any pair of adjacent levels it spans). Angelini et al. [1] showed that testing cCLP is NP-complete, but can be tested in quadratic time if the input graph is proper, thereby dropping the requirement of level-connectedness.

In this paper we consider two new variants that relax the conditions on the drawing of the cluster. In unrestricted Clustered Level Planarity (short uCLP) we keep the conditions (i)–(iii) as stated above, i.e., the shapes of clusters are not restricted by the levels. Our second variant y-monotone Clustered Level Planarity (short y-CLP) is based on the characterization that a planar drawing $\mathcal{G}$ of a graph G is clustered planar w.r.t. to a clustering T if and only if it is possible to insert a set of *augmentation edges* into $\mathcal{G}$ in a planar way such that each cluster becomes connected and no cycle formed by vertices of a cluster μ encloses a vertex not in μ in its interior [5]. In analogy to this, we define a level-planar drawing to be *y-cl-planar* if it satisfies these conditions but additionally the augmentation edges can be added as y-monotone curves. Figure 1 shows that cCLP, uCLP, y-CLP are indeed different problems. We are not aware of work that concerns uCLP or y-CLP.

We show that uCLP can be solved in polynomial time if the input graph is biconnected and has a single *source*, i.e., a vertex that does not have neighbors on a lower level. On the other hand we show that y-CLP is NP-complete under the same conditions and also if the number of levels is 5 and there is only a single non-trivial cluster, i.e., that not contains all vertices.

[1] they only considered this convex setting and used the name CLP instead of cCLP

2 Single-Source Biconnected (unrestricted) Clustered Level Planarity

We show that uCLP can be solved efficiently if G is a biconnected graph with a single source. To this end, we combine the polynomial-time solution for Clustered Planarity [2] with a combinatorial description of all level-planar drawings of a biconnected single-source graph [4].

Note that whether a drawing of G is clustered planar depends only on its combinatorial embedding, rather than the precise drawing. Thus, we call an embedding $\mathcal{E}$ of G *clustered planar* if the corresponding drawings are. We call an embedding $\mathcal{E}$ of G *level planar* if G admits a level-planar drawing with embedding $\mathcal{E}$. To solve uCLP for an instance (G, γ, T), we need to find an embedding of G that is both cluster planar and level planar.

We first introduce yet another type of constraints called *synchronized fixed-vertex constraints* (*sfv-constraints* for short). For a graph $G = (V, E)$ an *sfv-constraint* is a set Q of pairs (v, σ_v), where $v \in V$ and σ_v is a fixed cyclic order of the edges incident to v, called its *default rotation*. An embedding $\mathcal{E}$ of G *satisfies* the constraint Q if for each pair $(v, \sigma_v) \in Q$ the rotation of v in $\mathcal{E}$ is its default rotation or if for each pair $(v, \sigma_v) \in Q$ the rotation of v in $\mathcal{E}$ is the reverse of its default rotation. Given a set $\mathcal{Q}$ of sfv-constraints, we say that an embedding of G *satisfies* $\mathcal{Q}$ if it satisfies each $Q \in \mathcal{Q}$.

We use sfv-constraints to bridge from level-planarity to usual planarity. To this end, we introduce a slightly generalized version of Clustered Planarity, where we seek a clustered-planar embedding that additionally satisfies a given set $\mathcal{Q}$ of sfv-constraints. This problem is called Sync CP. The point is that the algorithm of Bläsius et al. [2] reduces Clustered Planarity to the intermediate problem Synchronized Planarity, which includes the option to directly express sfv-constraints. The reduction that shows the following lemma can be found in the full version of this paper at `arXiv:2402.13153` together with all further missing proofs.

▶ **Lemma 2.1.** Sync CP *can be solved in $O(n^3)$ time.*

We now turn to the second ingredient. Let (G, γ) be a biconnected single-source level-planar graph and let Γ be an arbitrary level-planar embedding of G. Brückner and Rutter [4] showed that there exists a data structure, called LP-tree, very similar to the famous SPQR-tree, that represents precisely the level-planar embeddings of G; see fig. 2 for an example. Like the SPQR-tree, the embeddings decisions for the LP-tree are made by (i) arbitrarily reordering parallel subgraphs between a pair of vertices and (ii) flipping the embeddings of some disjoint and otherwise rigid structures. Hence, the possible orderings of the edges around each vertex v of G in any level-planar embedding can be described by a PQ-tree [3, 7] T_v, called *level PQ-tree* that is straightforwardly derived from the LP-tree; it contains one P-node $u_{v,\mu}$ for each parallel structure μ in which v occurs and one Q-node $u_{v,\rho}$ for each rigid structure ρ, in which v occurs; see fig. 2a. The level-planar embedding Γ is used as reference to determine a *default rotation* $\sigma_{v,\rho}$ for each Q-node $u_{v,\rho}$.

If an embedding $\mathcal{E}$ of G is level-planar, the rotation of each vertex v is necessarily represented by its level PQ-tree T_v. It further holds that all Q-nodes $u_{v,\rho}$ with $v \in V$ that stem from the same rigid structure either all have their default orientation or all its reversal. An embedding where the last condition holds for each rigid structure is called *ρ-consistent*.

▶ **Lemma 2.2.** *An embedding $\mathcal{E}$ is level-planar if and only if the rotation of each vertex v is represented by its level PQ-tree T_v and moreover $\mathcal{E}$ is ρ-consistent.*

For a biconnected single-source level-planar graph G with level-planar embedding Γ, we derive a new graph G^+ and a set $\mathcal{Q}$ of synchronized fixed-vertex constraints. We replace

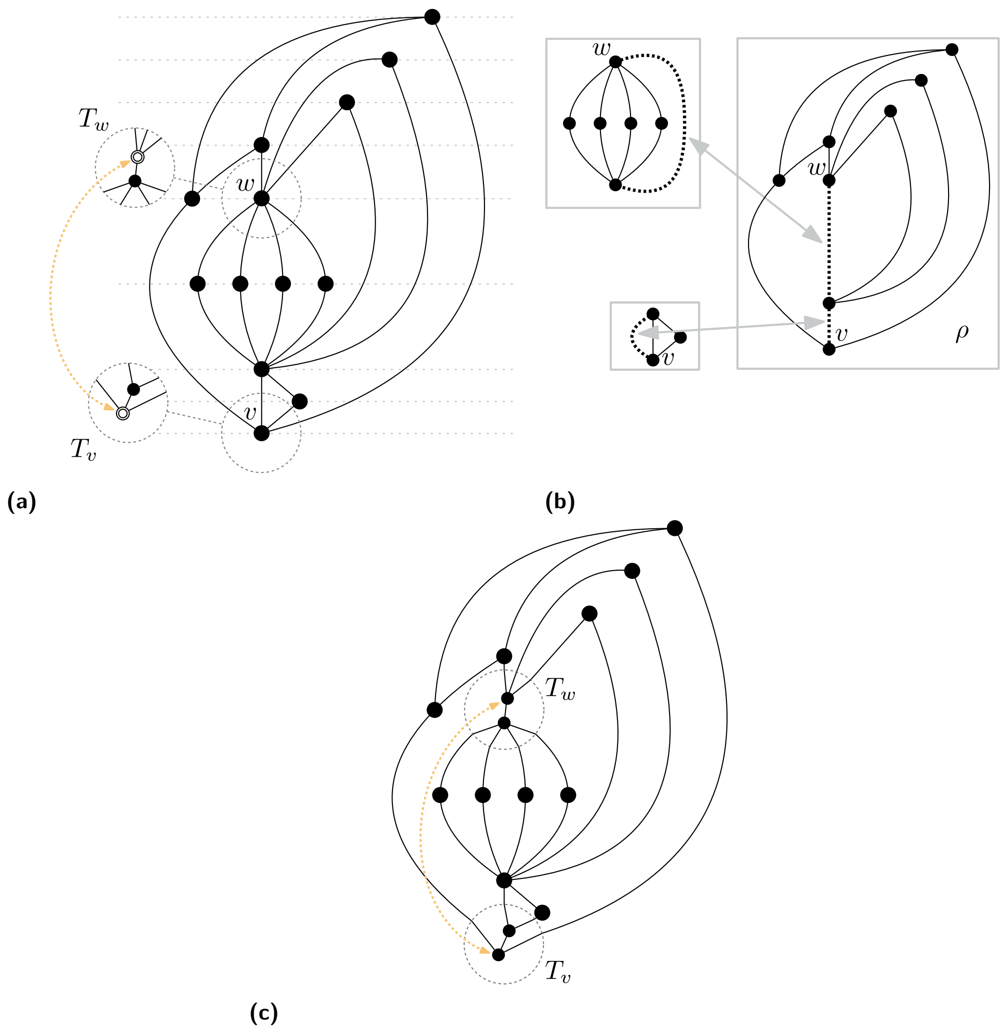

Figure 2 **(a)** A level graph G with two level PQ-trees T_w and T_v derived from **(b)** its LP-tree. P-nodes are represented by black disks, Q-nodes as white double disks. **(c)** The graph after replacing w, v by T_w, T_v; the orange arrow indicates the sfv-constraint due to ρ.

each vertex v of G by a tree isomorphic to its level PQ-tree T_v; see Figure 2c. In order to enforce ρ-consistency, we additionally create for each rigid structure ρ of the LP-tree an sfv-constraint $Q_\rho = \{(u_{v,\rho}, \sigma_{v,\rho}) \mid v \text{ occurs in the rigid structure } \rho\}$. Let $\mathcal{Q}$ denote the set of these constraints for all rigid structures. Clearly, we can obtain a planar embedding of G by taking a planar embedding of G^+ that satisfies $\mathcal{Q}$ and contracting each tree T_v back into a single vertex. The embeddings we can obtain in this way are precisely those where the rotation of each vertex v is represented by its level PQ-tree T_v and that are ρ-consistent.

Finally, it is time to connect clusters and level planarity. To this end, consider a clustering T on G. We naturally obtain a corresponding clustering T^+ of G^+ by placing each vertex of G^+ into the cluster of the vertex of G it replaces.

▶ **Lemma 2.3.** (G, γ, T) *admits a planar embedding that is level-planar and clustered-planar if and only if* (G^+, T^+) *admits a clustered-planar embedding that satisfies* $\mathcal{Q}$.

Altogether, this reduces the problem uCLP of biconnected single-source graphs to Sync CP, which can be solved efficiently by Lemma 2.1.

▶ **Theorem 2.4.** uCLP *can be solved in* $O(n^3)$ *time for biconnected single-source level graphs.*

3 Hardness of y-monotone Clustered Level Planarity

It is easy to see that y-CLP lies in NP by guessing and verifying the augmentation edges and an embedding. We show that it is NP-hard even for inputs with very restricted properties.

▶ **Theorem 3.1.** y-monotone Clustered Level Planarity *is NP-complete, even if the input is a biconnected graph with just one source.*

Proof Sketch. We reduce from the NP-complete problem Planar Monotone 3-SAT [6], which asks for the satisfiablity of 3-SAT formulas whose incidence graph has a drawing where all variables lie on a vertical line ℓ, each clause contains only positive or only negative literals, and the positive and negative clauses lie on opposing sides of ℓ; see Figure 3a.

Given such a drawing for a formula ϕ, we first reorient the clauses horizontally above the variables as illustrated in Figure 3b. From this drawing, we construct an equivalent instance of y-CLP as illustrated in Figure 3c. The variables and literals are represented by triconnected pillars (a $(3 \times k)$-grid for suitable k) that extend vertically towards the clause gadgets. The horizontal flip of a pillar represents its truth value. To synchronize pillars of the same variable, we use wedges that enclose a vertex of a disconnected cluster and, due to the required y-monotonicity for cluster connections, prohibit corresponding wedges of adjacent pillars to face each other, as such a connection would have to bypass both wedges. Using two clusters per variable, we can thus ensure consistent flips for all variables; see Figure 3c.

Using a similar approach with wedges, we can construct a clause gadget that allows all assignments for its literals except when all three literals are false; see Figure 3c for the structure of the gadget. As shown in Figure 4, there exists one configuration for the literal pillars of a clause where no valid embedding is possible, as the cluster of the gadget cannot connect with only y-monotone curves. Figure 5 shows valid embeddings for all other configurations. This way, we can construct in polynomial time an equivalent instance of y-CLP where the graph is biconnected and only has a single source. ◀

Our second reduction is from 3-Partition, whose input is a multiset $A = \{a_1, \ldots, a_{3m}\}$ positive integers and a bound $B \in \mathbb{N}^+$ with $B/4 < a_i < B/2$ and $\sum_{a \in A} a = m \cdot B$.

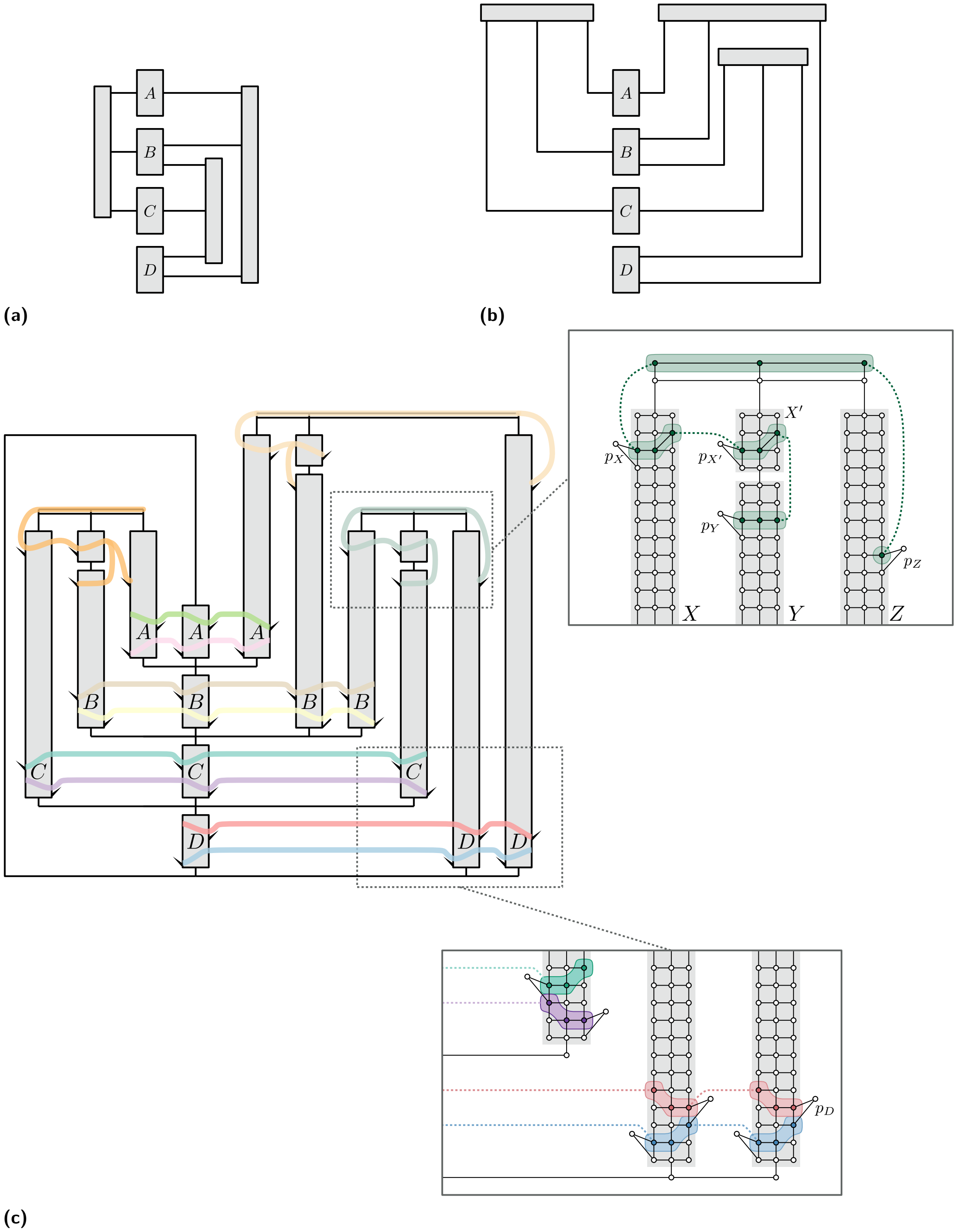

Figure 3 **(a)** An instance of PLANAR MONOTONE 3-SAT. **(b)** The modified incidence graph. **(c)** The structure of the corresponding y-CLP instance. Highlighted are a clause gadget (top right) and the gadget for propagating variable assignments (bottom right).

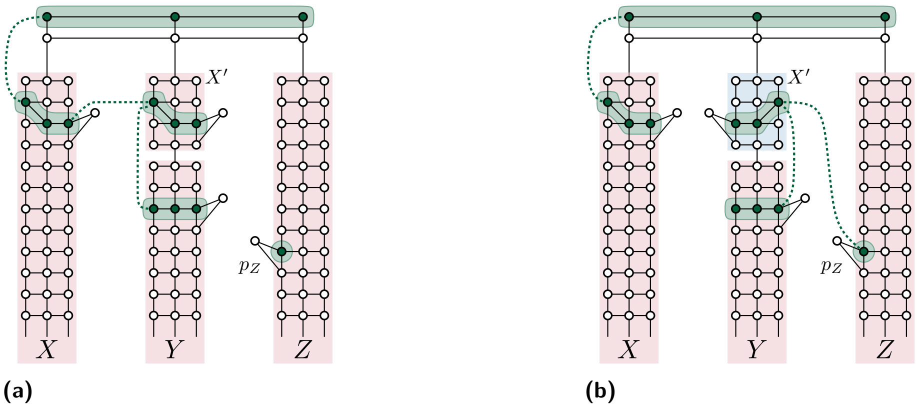

Figure 4 Neither flip of X' admits a valid embedding of the clause gadget if all literals are false.

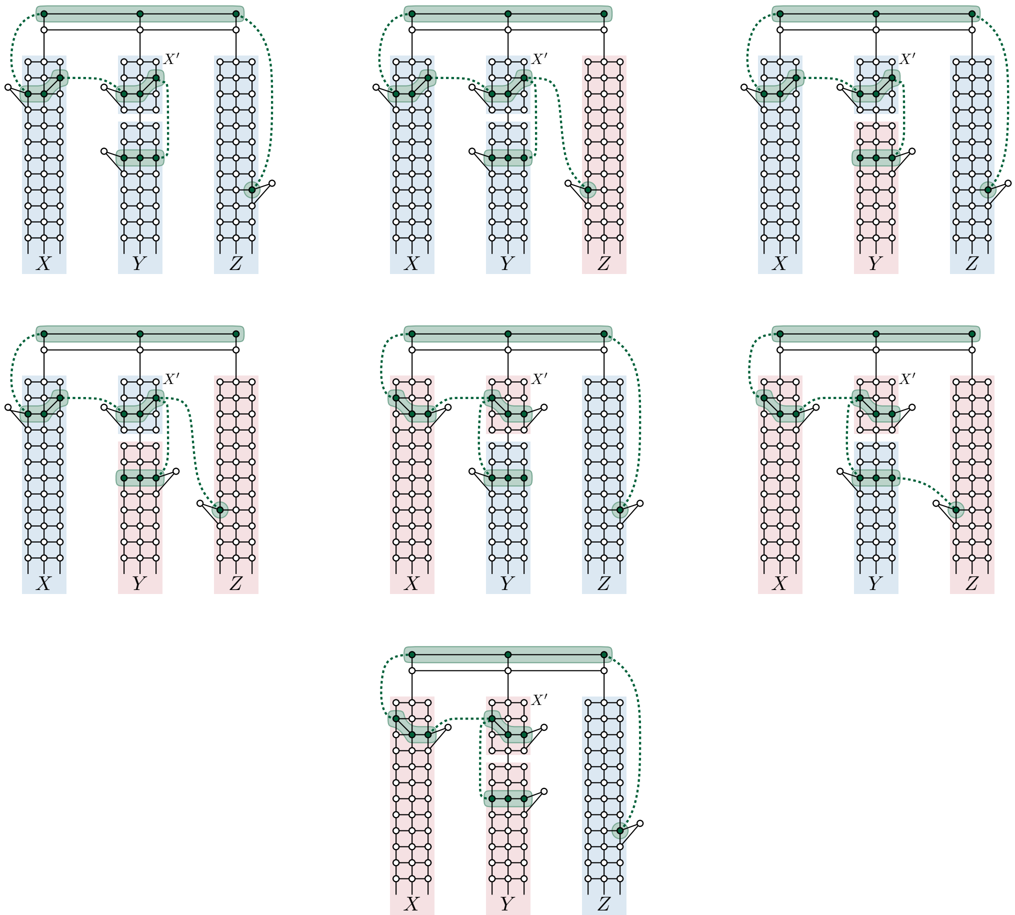

Figure 5 The seven variable assignments for which the clause gadget admits a valid embedding.

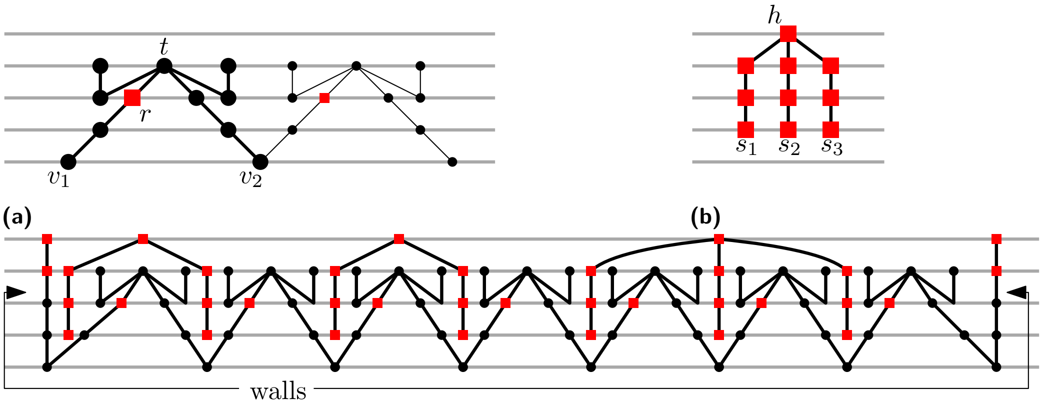

Figure 6 **(a)** A receiver (bold) with a marked vertex r (box), and a second receiver (non-bold) chained to the first one. **(b)** A 3-plug. **(c)** A bucket of size 7, filled with two 2-plugs and a 3-plug. Every marked bucket vertex can be connected to a pin with a y-monotone curve.

The question is whether A can be partitioned into m sets $A_1, A_2, \ldots, A_m$, such that for every $j \in \{1, \ldots, m\}$ it is $\sum_{a \in A_j} a = B$. 3-Partition is strongly NP-complete, i.e., it remains NP-complete even if B is polynomial in m [10].

▶ **Theorem 3.2.** *y-monotone Clustered Level Planarity is NP-complete, even if the input contains only one non-trivial cluster, the number of levels is at most 5, and all vertices have a fixed rotation.*

Proof Sketch. Let (m, A, B) be an instance of 3-Partition. We construct an instance of uCLP with a single non-trivial cluster μ. The main idea is to build m *buckets* (the structure in Figure 6c) by chaining B *receivers* (the structure in Figure 6a, each of which contains a *connector vertex* r that belongs to μ), and closing the sides of each bucket with two paths (also called *walls*). Note that each of the B connector vertices of a bucket must be connected to the rest of μ due to the paths of length 2 attached to the vertices marked as t in the receiver; see Figure 6a. For every $a \in A$, we generate an *a-plug*; the structure in Figure 6b). The leaves of a plug are called *pins* and these are the only vertices that can link the connector vertices to the remainder of cluster μ. Given a plug and a bucket, either all of the vertices of the plug are drawn between the two bucket walls, or none of them. Thus, we can model a solution $A_1, \ldots, A_m$ with the m buckets, and a drawing assigns $a \in A$ to A_i if and only if the corresponding a-plug is in the i-th bucket. Since every pin can join at most one connector vertex, there are at least B pins inside a bucket in a valid drawing. Since there are m buckets and a total of $m \cdot B$ pins, the instance is valid if and only if we can distribute the plugs in such a way that there are precisely B pins per bucket, which corresponds directly to a solution of (m, A, B). ◀

4 Conclusion

We have introduced the problems unrestricted Clustered Level Planarity and y-monotone Clustered Level Planarity, gave an polynomial-time algorithm for unrestricted Clustered Level Planarity if restricted to biconnected single-source graphs, and showed that y-monotone Clustered Level Planarity is NP-complete under very restricted conditions.

We conclude by providing some open questions. On the one hand, these are inspired by the restrictions imposed by our algorithm. The LP-trees we use in Theorem 2.4 only exist for biconnected single-source instances and it is unlikely that this concept can be extended to multiple sources [4, Section 5]. Is it possible to extend our algorithm to non-biconnected graphs? More generally, what is the complexity of unrestricted Clustered Level Planarity? On the other hand, the NP-hardness results on y-monotone Clustered Level Planarity and (non-proper) Convex Clustered Level Planarity [1] raise the question whether these problems are FPT with respect to natural parameters.

References

1 Patrizio Angelini, Giordano Da Lozzo, Giuseppe Di Battista, Fabrizio Frati, and Vincenzo Roselli. The importance of being proper: (in clustered-level planarity and t-level planarity). *Theoretical Computer Science*, 571:1–9, 2015. doi:10.1016/j.tcs.2014.12.019.

2 Thomas Bläsius, Simon D. Fink, and Ignaz Rutter. Synchronized planarity with applications to constrained planarity problems. *ACM Transactions on Algorithms*, 2023. doi:10.1145/3607474.

3 Kellogg S. Booth and George S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *Journal of Computer and System Sciences*, 13(3):335–379, 1976. doi:10.1016/S0022-0000(76)80045-1.

4 Guido Brückner and Ignaz Rutter. An SPQR-tree-like embedding representation for level planarity. *Journal of Computational Geometry*, 14(1):48–77, 2023. doi:10.20382/JOCG.V14I1A3.

5 Sabine Cornelsen and Dorothea Wagner. Completely connected clustered graphs. *Journal of Discrete Algorithms*, 4(2):313–323, 2006. doi:10.1016/j.jda.2005.06.002.

6 Mark de Berg and Amirali Khosravi. Optimal binary space partitions for segments in the plane. *Int. J. Comput. Geom. Appl.*, 22(3):187–206, 2012. doi:10.1142/S0218195912500045.

7 Simon D. Fink, Matthias Pfretzschner, and Ignaz Rutter. Experimental comparison of PC-trees and PQ-trees. *ACM Journal of Experimental Algorithmics*, 28(1):10:1–10:24, 2023. doi:10.1145/3611653.

8 Michael Forster and Christian Bachmaier. Clustered level planarity. In Peter van Emde Boas, Jaroslav Pokorný, Mária Bieliková, and Julius Stuller, editors, *Proceedings of the 30th Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM'04)*, volume 2932 of *Lecture Notes in Computer Science*, pages 218–228. Springer, 2004. doi:10.1007/978-3-540-24618-3_18.

9 Radoslav Fulek and Csaba D. Tóth. Atomic embeddability, clustered planarity, and thickenability. *Journal of the ACM*, 69(2):13:1–13:34, 2022. arXiv:1907.13086v1, doi:10.1145/3502264.

10 Michael R. Garey and David S. Johnson. Complexity results for multiprocessor scheduling under resource constraints. *SIAM Journal on Computing*, 4(4):397–411, 1975. doi:10.1137/0204035.

11 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.