Level Planarity Is More Difficult Than We Thought

Simon D. Fink
Algorithms and Complexity Group, Technische Universität Wien, Austria

Matthias Pfretzschner
Faculty of Computer Science and Mathematics, Universität Passau, Germany

Ignaz Rutter
Faculty of Computer Science and Mathematics, Universität Passau, Germany

Peter Stumpf
Charles University, Prague, Czech Republic

Abstract

We consider three simple quadratic-time algorithms for Level Planarity and give a level-planar instance that they either falsely classify as negative or for which they output a non-planar drawing.

2012 ACM Subject Classification Human-centered computing → Graph drawings

Keywords and phrases level planarity, 2-SAT, simple algorithm, counterexample

Digital Object Identifier 10.4230/LIPIcs.GD.2024.50

Category Poster Abstract

Related Version *Full Version including Poster*: `https://arxiv.org/abs/2409.01727`

Funding *Simon D. Fink*: Vienna Science and Technology Fund (WWTF) [10.47379/ICT22029].
Peter Stumpf: Czech Science Foundation grant GAČR 23-04949X.

1 Introduction

Given a graph $G = (V, E)$ and a *level assignment* $\ell : V(G) \to \mathbb{N}$, the problem Level Planarity asks for a crossing-free drawing of G where vertices have their prescribed level as y-coordinate and all edges are y-monotone. When initially considering the problem in 1988, Di Battista and Nardelli [1] gave a linear-time algorithm for the restricted case where the graph is a hierarchy, i.e., only one vertex has no neighbors on a lower level. A subsequent attempt to extend this algorithm to the general case [7] was shown to be incomplete [10]. Jünger et al. finally gave the first linear-time algorithm for testing [11] and embedding [8, 9] level graphs around the turn of the millennium. Because this algorithm is quite involved, slower but simpler algorithms were developed by Randerath et al. [12], Healy and Kuusik [6], as well as Harrigan and Healy [5] in the decade thereafter. All these algorithms consider the pairwise ordering of vertices on the same level, greedily fixing an order for a (certain) pair and then checking for further orders implied by this. If the process terminates without finding a contradiction, we obtain a total vertex order for each level and thereby a level planar embedding. In the following, we give a level-planar counterexample that each known variant of this algorithm either incorrectly classifies as negative instance or correctly identifies as positive instance but outputs a drawing that is not planar. To the best of our knowledge, this leaves no correct *simple* embedding algorithm for level graphs. In particular, we are not aware of *any* correct implementation for embedding level-planar graphs.

Randerath et al. use an explicit 2-SAT formulation for the pairwise orders of vertices on the same level. Due to known gaps in the proof of Randerath et al., Brückner et al. [2, 3] showed this characterization via a 2-SAT formula is equivalent to the Hanani-Tutte-style characterization of Level Planarity [4]. Thereby, our counterexample only breaks the proof of correctness as well as the embedder by Randerath et al., while their 2-SAT formulation still yields a correct *test* for Level Planarity via this indirect proof [2, 3].

© Simon D. Fink, Matthias Pfretzschner, Ignaz Rutter, and Peter Stumpf;
licensed under Creative Commons License CC-BY 4.0
32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).
Editors: Stefan Felsner and Karsten Klein; Article No. 50; pp. 50:1–50:3
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

2 Randerath et al.

The algorithm by Randerath et al. [12] works as follows. First, edges spanning multiple levels are subdivided such that subsequently edges only occur between adjacent levels, resulting in a *proper* level graph. The planarity of the resulting graph is then tested using a 2-SAT formula. The formula contains a variable $(a < b)$ for every pair a, b of vertices that appear on the same level, encoding the relative order of these two vertices. For every pair of edges $uv, xy \in E$ with $\ell(u) = \ell(x) = \ell(v) + 1 = \ell(y) + 1$ with $u \neq x, v \neq y$ it adds the 2-SAT constraint $(u < x) \Leftrightarrow (v < y)$. Combining this with the constraints for antisymmetry $((a < b) \Leftrightarrow \neg(b < a))$ and transitivity $((a < b) \wedge (b < c) \Rightarrow (a < c))$ necessary for finding total orders yields a 3-SAT formula. However, Randerath et al. [12] show that omitting the transitivity constraints yields an *equisatisfiable* 2-SAT formula. To prove this equivalence, they show that the 2-SAT formula can be used to compute a level-planar embedding of the input graph. They greedily pick and assign equivalence classes of the formula in arbitrary order, but prioritize transitive closures where possible. Figure 1 shows a counterexample where the algorithm gives a false-negative answer when assigning classes in the shown order.

3 Healy and Kuusik & Harrigan and Healy

The algorithms by Healy and Kuusik [6] as well as the one by Harrigan and Healy [5] uses a similar concept. Instead of working with equivalence classes of a 2-SAT formula, they work with connected components of the closely related *vertex exchange graph* (ve-graph). This graph contains one vertex for every ordered pair of vertices that appear on the same level. Two vertices of the ve-graph are adjacent if they correspond to a pair of independent edges between the same levels. Starting with an arbitrary drawing $\mathcal{L}$ of the input graph, the edges of the ve-graph are first labeled with $+$ or $-$, depending on whether the corresponding edges cross in $\mathcal{L}$. Subsequently, a DFS is used to test the ve-graph for odd-labeled cycles, which corresponds to a contradiction within a 2-SAT equivalence class. The two algorithms now differ slightly in how they continue to construct an embedding. Similar to Randerath et al., Healy and Kuusik [6] fix the orders of vertex pairs (i.e., whether all pairs of a connected ve-graph component are swapped or not) in an arbitrary order, also performing the transitive closure if possible. Thus, the processing order from Figure 1 also breaks this approach.

The later Harrigan and Healy approach [5] is slightly more involved. During the DFS traversal, they already change the relative order of some vertex pairs compared to the initial drawing $\mathcal{L}$ [5, Algorithm 1]. Subsequently, the ve-graph is traversed again in a specific order and, for some vertices of $\mathcal{L}$, the chosen vertex order is flipped [5, Algorithm 2]. Using choices as shown in Figure 2, this does not yield a planar embedding even for a positive instance.

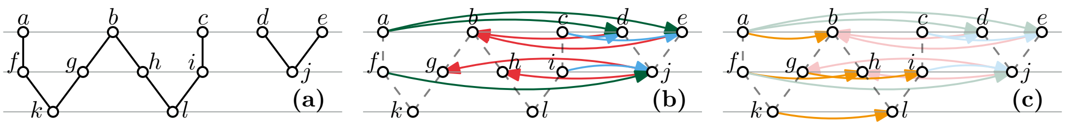

Figure 1 (a) A level-planar graph G. **(b)** The green, blue, and red 2-SAT equivalence classes can be greedily assigned in this order. Subsequently, transitive closure forces $a < b$ as well as $i < g$, but the planarity constraints force $a < b \leftrightarrow f < h \leftrightarrow k < l \leftrightarrow g < i$ **(c)**, yielding a contradiction.

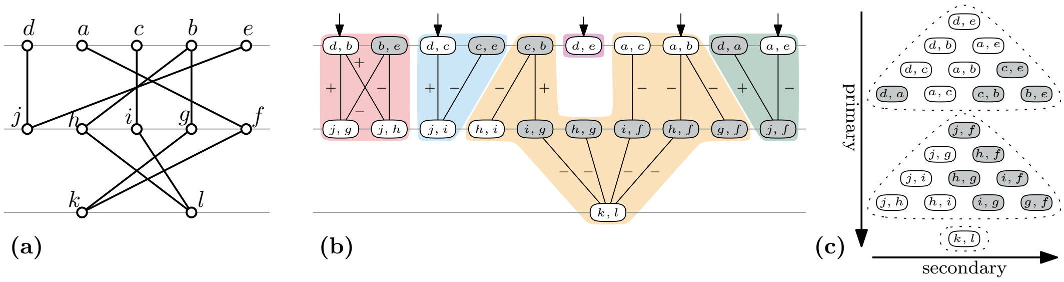

Figure 2 **(a)** An initial drawing $\mathcal{L}$ for Figure 1a. **(b)** The corresponding labeled ve-graph. Arrows mark the chosen DFS entry points, pairs marked as swapped by Algorithm 1 are shown in gray. **(c)** The processing order for the vertices of the ve-graph in Algorithm 2.

References

1. Giuseppe Di Battista and Enrico Nardelli. Hierarchies and planarity theory. *IEEE Transactions on Systems, Man and Cybernetics*, 18(6):1035–1046, 1988. `doi:10.1109/21.23105`.
2. Guido Brückner, Ignaz Rutter, and Peter Stumpf. Level planarity: Transitivity vs. even crossings. In Therese Biedl and Andreas Kerren, editors, *Proceedings of the 26th International Symposium on Graph Drawing and Network Visualization (GD '18)*, volume 11282 of *Lecture Notes in Computer Science*, pages 39–52. Springer, 2018. `doi:10.1007/978-3-030-04414-5_3`.
3. Guido Brückner, Ignaz Rutter, and Peter Stumpf. Level-planarity: Transitivity vs. even crossings. *Electronic Journal of Combinatorics*, 29(4), 2022. `doi:10.37236/10814`.
4. Radoslav Fulek, Michael J. Pelsmajer, Marcus Schaefer, and Daniel Štefankovič. Hanani-Tutte, monotone drawings, and level-planarity. In János Pach, editor, *Thirty Essays on Geometric Graph Theory*, pages 263–287. Springer, 2013. `doi:10.1007/978-1-4614-0110-0_14`.
5. Martin Harrigan and Patrick Healy. Practical level planarity testing and layout with embedding constraints. In Seok-Hee Hong, Takao Nishizeki, and Wu Quan, editors, *Proceedings of the 15th International Symposium on Graph Drawing (GD '07)*, volume 4875 of *Lecture Notes in Computer Science*, pages 62–68. Springer, 2007. `doi:10.1007/978-3-540-77537-9_9`.
6. Patrick Healy and Ago Kuusik. Algorithms for multi-level graph planarity testing and layout. *Theoretical Computer Science*, 320(2-3):331–344, 2004. `doi:10.1016/J.TCS.2004.02.033`.
7. Lenwood S. Heath and Sriram V. Pemmaraju. Recognizing leveled-planar dags in linear time. In Franz-Josef Brandenburg, editor, *Proceedings of the 3rd International Symposium on Graph Drawing (GD '95)*, volume 1027 of *Lecture Notes in Computer Science*, pages 300–311. Springer, 1995. `doi:10.1007/BFB0021813`.
8. Michael Jünger and Sebastian Leipert. Level planar embedding in linear time. In Jan Kratochvíl, editor, *Proceedings of the 7th International Symposium on Graph Drawing (GD '99)*, volume 1731 of *Lecture Notes in Computer Science*, pages 72–81. Springer, 1999. `doi:10.1007/3-540-46648-7_7`.
9. Michael Jünger and Sebastian Leipert. Level planar embedding in linear time. *Journal of Graph Algorithms and Applications*, 6(1):67–113, 2002. `doi:10.7155/JGAA.00045`.
10. Michael Jünger, Sebastian Leipert, and Petra Mutzel. Pitfalls of using pq-trees in automatic graph drawing. In Giuseppe Di Battista, editor, *Proceedings of the 5th International Symposium on Graph Drawing (GD '97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 193–204. Springer, 1997. `doi:10.1007/3-540-63938-1_62`.
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`.
12. 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. *Electronic Notes Discreete Mathematics*, 9:269–277, 2001. `doi:10.1016/S1571-0653(04)00327-0`.