Proper Level Graph Embedding

Maciej Brzozowski

June 2025

1 Introduction

Given a graph $G = (V, E)$ and a level assignment $l : V(G) \to \mathbb{N}$, the Level Planarity problem asks for a planar drawing of a graph such that y-coordinate of a vertex a is $l(a)$.
For Proper Level graphs l also satisfies

$$(x, y) \in E \Rightarrow |l(x) - l(y)| = 1$$

(edges are only between adjacent levels)

Jüngen et al. presented a linear time algorithm for Level Planarity problem [5]. Due to the fact that this algorithm is quite complicated, there were attempts to develop simpler (but slower) algorithms ([3],[4],[7]) for the proper level graphs, but recently Fink et al.[2] provided a counterexample that disproved the correctness of those simple embedders.

I present an $O(n^4)$ algorithm that produces the correct embedding of a proper level graphs. The implementation of the algorithm is publicly available at `http://github.com/Ambiguouss/level-planar-drawing`.

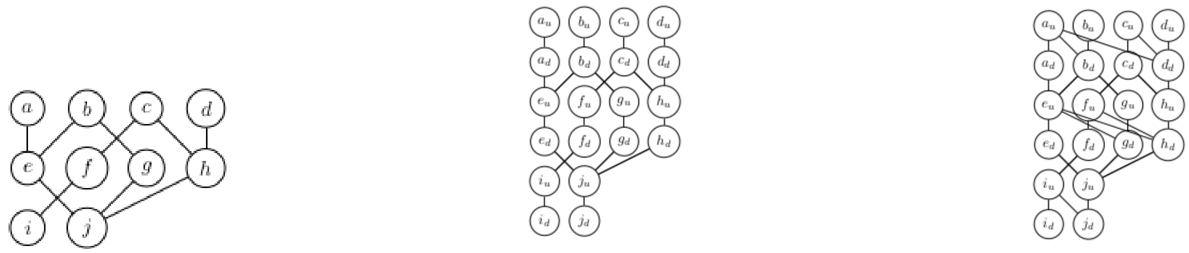
2 Algorithm

Assume that we have an oracle that for a given proper level graph answers whether a graph is level planar. Linear algorithm was presented by Jüngen et al. [6], but I decided on using a slower, but simpler algorithm by Randerath et al. [7] that works in $O(n^2)$ time. Note that the embedding algorithm described by Randerath et al. fails on the counterexample provided by Fink et al., but the algorithm for deciding whether a graph is level planar is correct due to the Hannani-Tutte style proof by Brückner et al.[1].

Let H be a proper level graph created from a proper level graph G in the following way. For every level k, create two levels: k_{up} and k_{down}. For every vertex $v \in k$, create two vertices $v_{up} \in k_{up}$ and $v_{down} \in k_{down}$. For every vertex x, add an edge (x_{up}, x_{down}). For every (x, y) edge, such that $l(x) = l(y)+1$ add (x_{down}, y_{up}) edge. Notice that H is level planar iff G is level planar. Now we will add some edges to H following this procedure:

$R = H$
for $x \in G, y \in G$ such that $l(x) = l(y)$ **do**
  if $R \cup (x_{up}, y_{down})$ is level planar **then**
    $R = R \cup (x_{up}, y_{down})$

Notice that R is level planar iff H is level planar.



(a) G (b) H (c) R

Figure 1

For $v \in V(R)$ Let $D(v) = \{y \in V(R) : l(y) = l(x) - 1 \wedge (v, y) \in E(R)\}$ (set of neighbors on a lower level). Similarly let $U(v) = \{y \in V(R) : l(y) = l(x) + 1 \wedge (v, y) \in E(R)\}$ (set of neighbors on a higher level).
For $x \in G$ Let $N(x) = (D(x_{up}) \cup U(x_{down})) \setminus \{x_{up}, x_{down}\}$.

Consider a planar embedding of R. Notice, that for every level, for the left (or right) most vertex x in this embedding we have $|N(x)| = 1$ (assuming that k has more than one vertex). Additionally, if $|N(y)| = 1$, then y must be left (or right) most vertex in this embedding. Also, for every vertex in x, $|N(x)| \leq 2$.
It's easy now to determine the vertex ordering on each level (up to reversal):

for level $k \in G$ **do**
  $res_k = []$
  $visited[a] = 0$ for every vertex $a \in k$
  Find x, such that $|N(x)| = 1$ in R
  res_k.push_back(x)
  $visited[x] = 1$
  do
    Find $y \in N(x)$ such that $visited[y] == 0$
    res_k.push_back(y)
    $visited[y] = 1$
    $x = y$
  while $|N(x)| == 2$

(a) R in order (b) Planar Embedding of G

Figure 2

Now we have an order on every level, but we also need to check if we need to reverse it. Notice that we do not have to reverse the first level (reflection of a planar embedding is also a planar embedding). For level k, we just check up neighbors of the left most and right most vertices. If the up neighbors of the right most vertex is to the left of the neighbors of the left most vertex then we have to reverse the order.

Note that the construction of R in $O(n^2 \cdot C(n))$, where $C(n)$ is the complexity of checking level planarity, which yields $O(n^4)$ when using an algorithm by Randerath et al. [7]

References

[1] Guido Brückner, Ignaz Rutter, and Peter Stumpf. "Level-Planarity: Transitivity vs. Even Crossings". In: *Electronic Journal of Combinatorics* 29.4 (2022). DOI: 10.37236/10814.

[2] Simon D. Fink et al. *Level Planarity Is More Difficult Than We Thought*. 2024. arXiv: 2409.01727 [cs.DM]. URL: https://arxiv.org/abs/2409.01727.

[3] Martin Harrigan and Patrick Healy. "Practical Level Planarity Testing and Layout with Embedding Constraints". In: *Graph Drawing 2007*. Ed. by Seok-Hee Hong, Takao Nishizeki, and Wu Quan. Vol. 4875. Lecture Notes in Computer Science. Springer, 2007, pp. 62–68. DOI: 10.1007/978-3-540-77537-9_9.

[4] Patrick Healy and Ago Kuusik. "Algorithms for multi-level graph planarity testing and layout". In: *Theoretical Computer Science* 320.2 (2004), pp. 331–344. ISSN: 0304-3975. DOI: https://doi.org/10.1016/j.tcs.2004.02.033.

[5] Michael Jünger and Sebastian Leipert. "Level planar embedding in linear time". In: *J. Graph Algorithms Appl.* 6.1 (2002), pp. 67–113. DOI: https://doi.org/10.7155/jgaa.00045.

[6] Michael Jünger, Sebastian Leipert, and Petra Mutzel. "Level Planarity Testing in Linear Time". In: *Graph Drawing 1998*. Ed. by Sue Whitesides. Vol. 1547. Lecture Notes in Computer Science. Springer, 1998, pp. 224–237. DOI: 10.1007/3-540-37623-2_17.

[7] Bert Randerath et al. "A Satisfiability Formulation of Problems on Level Graphs". In: *Electronic Notes in Discrete Mathematics* 9 (2001). LICS 2001 Workshop on Theory and Applications of Satisfiability Testing (SAT 2001), pp. 269–277. ISSN: 1571-0653. DOI: https://doi.org/10.1016/S1571-0653(04)00327-0.