# Track Planarity Testing and Embedding*

Christian Bachmaier, Franz J. Brandenburg, and Michael Forster

University of Passau, 94030 Passau, Germany
{bachmaier,brandenb,forster}@fmi.uni-passau.de

**Abstract.** A track graph is a graph with its vertex set partitioned into horizontal levels. It is track planar if there are permutations of the vertices on each level such that all edges can be drawn as weak monotone curves without crossings. The novelty and generalisation over level planar graphs is that horizontal edges connecting consecutive vertices on the same level are allowed. We show that track planarity can be reduced to level planarity in linear time. Hence, there are $\mathcal{O}(|V|)$ time algorithms for the track planarity test and for the computation of a track planar embedding.

## 1 Introduction

In automated graph drawing one goal is to display graphs with a given hierarchical ordering of its vertices nicely, e. g., hierarchical and planar. A particular graph class for which this is possible are the so called level planar graphs investigated in [5,8,13,14,16,17,19–22]. To obtain an even larger class of hierarchical graphs for which hierarchical planar drawings can be generated efficiently we introduce track graphs where edges connecting consecutive vertices on the same hierarchical level are allowed.

This is a generalisation of level graphs which is similar to the extension of strictly upward drawings of binary trees to upward drawings. In strictly upward drawings the edges are strictly monotone, whereas in upward drawings also horizontal edges are allowed. It is well-known [7] that $\Theta(|V| \log |V|)$ is the upper and lower bound for the area of strictly upward drawings of binary trees while only $\mathcal{O}(|V|)$ area is needed for upward drawings, see [11].

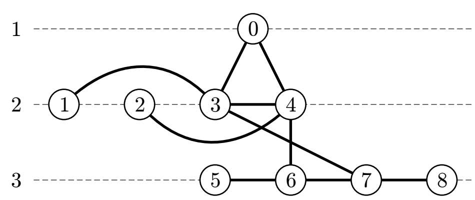
As our main result we show that the test on track planarity and the computation of a track planar embedding, which preserves the hierarchical ordering of the vertices, can be done in linear time. This is done by a linear time reduction to level planarity.

A *k-level graph* $G = (V, E, \phi)$ is an undirected graph with a level assignment $\phi\colon V \to \{1, 2, \ldots, k\}$, $1 \leq k \leq |V|$, that partitions the vertex set into $V = V^1 \mathbin{\dot\cup} V^2 \mathbin{\dot\cup} \cdots \mathbin{\dot\cup} V^k$, $V^i = \phi^{-1}(i)$, $1 \leq i \leq k$, such that $\phi(u) \neq \phi(v)$ for each edge $(u, v) \in E$. A $k$-level graph is *proper* if $|\phi(u) - \phi(v)| = 1$ for each edge $(u, v) \in E$. It is $k$-level planar if it is possible to draw it in the Cartesian plane

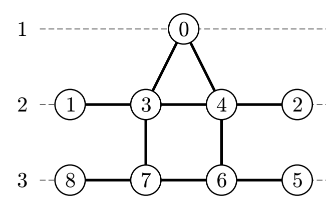
* This research has been supported in part by the Deutsche Forschungsgemeinschaft, grant BR 835/9-1.

such that all vertices $v \in V^i$ on the $i$-th level are placed on a single horizontal line $l_i = \{ (x, i) \mid x \in \mathbb{R} \}$ and the edges are drawn as vertically strictly monotone curves without crossings. In order to obtain a $k$-level drawing with the above restrictions, an embedding has to be computed. Level planar embeddings are explicitly characterized by linear orderings $\leq_i$ of the vertices in each $V^i$.

A *$k$-track graph* $G = (V, E \cup \overline{E}, \phi)$ is a $k$-level graph with additional edges $\overline{E} \subseteq \{ (u, v) \mid u, v \in V, \phi(u) = \phi(v) \}$ within the same level (*track*[1]). It is *$k$-track planar* if there are linear orderings $\leq_i$, $1 \leq i \leq k$ of the vertices on each level such that edges are drawn as weak monotone curves without edge or vertex crossings. Thus in a track planar embedding $\mathcal{E}_l = (\leq_i)_{1 \leq i \leq k}$ all edges $(u, v) \in \overline{E}$ connect consecutive vertices on the same level $i = \phi(u)$, i. e., $u \leq_i v$ implies $\forall t \in V^i - \{u, v\}\colon \neg(u \leq_i t \leq_i v)$. Figure 1 shows a 3-track planar graph and a track planar drawing of it. Note that an additional edge $(6, 8)$ is not allowed.




(a) A 3-track planar graph




(b) A 3-track planar drawing of Fig. 1(a)

**Fig. 1.** A track planar graph and a track planar drawing of it

We do not consider the problem of finding a level planar or track planar embedding for graphs without a given levelling. Heath and Rosenberg show in [18] that the levelling problem is NP-hard for proper graphs. In the non-proper case it is trivial using a modification of the algorithm of de Fraysseix et al. [6]. This does not minimise the number of levels, however.

Without loss of generality we consider only simple graphs without self loops and parallel edges. Because of the following Lemma we reject simple input graphs with $|E| > 3|V| - 6$ as not level planar.

**Lemma 1.** *Let $G$ be a $k$-track graph. Then the following holds:*

$$G \textit{ is } k\textit{-level planar } \Rightarrow G \textit{ is } k\textit{-track planar } \Rightarrow G \textit{ is planar.}$$

## 2 Related Work

There are different approaches for solving the level planarity testing and embedding problem. Leipert et al. [19–22] present a linear time algorithm, based

[1] The term “track” derives from [10].

on previous work of Heath and Pemmaraju [16, 17]. As in the vertex addition method for planarity testing [9,23], they make use of the *PQ-tree* data structure introduced by Booth and Lueker [4]. This algorithm recognises level planarity of arbitrary level graphs in contrast to the previous algorithm of Di Battista and Nardelli [8], which is restricted to level graphs with a single source, so called *hierarchies*, or the algorithm of Chandramouli and Diwan [5] for triconnected level graphs.

Because Leipert's algorithm is rather difficult to understand and to implement, Healy and Kuusik [14] present a much simpler approach for the detection of level planarity. Their algorithm runs in $\mathcal{O}(|V|^2)$ time for proper level graphs and $\mathcal{O}(|V|^4)$ time in the general case. If an embedding is needed, the time complexities raise to $\mathcal{O}(|V|^3)$ and $\mathcal{O}(|V|^6)$, respectively.

## 3 Reduction to Level Planarity

Our basic idea for solving the track planarity problem is to reduce it to level planarity. Therefore we transform the track graph $G = (V, E, \phi)$ into a level graph $G' = (V', E', \phi')$ such that $G'$ is level planar if and only if $G$ is track planar. After initialising $V'$ with $V$ and $E'$ with $E$ we triple the number of levels by defining $\phi'(v) = 3\phi(v) - 1$ for all $v \in V'$. Then every *horizontal edge* $e = (u, v) \in E'$, with $\phi'(u) = \phi'(v)$ is replaced by a *diamond* subgraph, see Fig. 2. Two new vertices $v_e$ and $v'_e$ on the two adjacent levels are introduced and are connected to both end vertices of $e$. Afterwards $e$ is removed. We obtain $|V'| \in \mathcal{O}(|V|)$ and $|E'| \in \mathcal{O}(|E|)$.

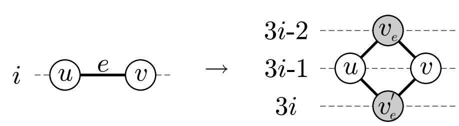



**Fig. 2.** Transformation of the horizontal edges into *diamonds*

*Example 1.* Figure 3 shows the transformation of a 3-track graph into an equivalent 9-level graph which contains two *diamond chains*. Note that the new vertices are on levels which do not contain original vertices and vice versa.

**Lemma 2.** *Let $G$ be a $k$-track graph and $G'$ its transformation. Then the following holds:*

$$G \textit{ is } k\textit{-track planar} \Leftrightarrow G' \textit{ is } 3k\textit{-level planar.}$$

*Proof.* Since isolated vertices affect neither track nor level planarity, we consider only graphs not containing isolated vertices. To prove the equivalence we prove two implications:

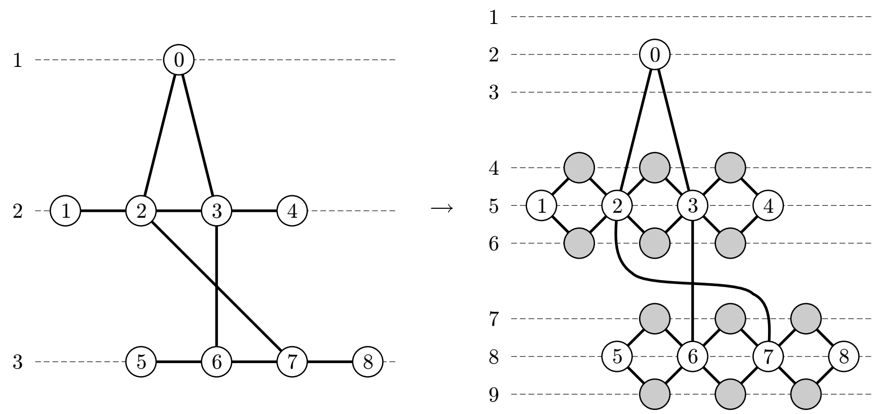



**Fig. 3.** Example of a track graph transformed into a level graph

"⇒": Let $G = (V, E, \phi)$ be $k$-track planar. Thus there exists a $k$-track planar embedding $\mathcal{E}_l$ of $G$. Now we construct an embedding $\mathcal{E}'_l$ of $G' = (V', E', \phi')$ from $\mathcal{E}_l$ by tripling the number of levels and by defining $\leq'_{3i-1} = \leq_i$, $1 \leq i \leq k$. The remaining relations $\leq'_{3i-2}$ and $\leq'_{3i}$ are given by ordering the dummy vertices on levels $3i-2$ and $3i$ according to their adjacent non-dummy vertices. The two new vertices of diamonds are always placed with respect to horizontal ordering between the end vertices of the corresponding horizontal edge $e$ without violating planarity. Suppose that $\mathcal{E}'_l$ is not level planar. Thus at least two edges $e_1$ and $e_2$ cross in $\mathcal{E}'_l$. If they are both non-diamond edges, they also cross in $\mathcal{E}_l$, a contradiction to the track planarity of $\mathcal{E}_l$. If exactly one of them is a diamond edge, suppose $e_2$, then $e_1$ crosses the horizontal edge in $\mathcal{E}_l$ due to which $e_2$ was introduced. Again a contradiction to the track planarity of $\mathcal{E}_l$. If both edges are diamond edges, we obtain again a contradiction because their corresponding horizontal edges overlap in $\mathcal{E}_l$ which is not allowed for track planar embeddings. Since $\mathcal{E}'_l$ is level planar, $G'$ is level planar, too.

"⇐": Let $G'$ be $3k$-level planar. Thus $G'$ has a $3k$-level planar embedding $\mathcal{E}'_l$. In $\mathcal{E}'_l$ the inner face of every diamond is empty. Apart from isolated vertices, which don't exist at this step, no vertex can be inside without violating level planarity. Thus it is possible to draw an edge in $G$ between the two original vertices of every diamond without violating planarity. Every level $i$ with $i \not\equiv 2 \pmod 3$ and all vertices on level $i$ together with their adjacent edges can be deleted. Deletions never hurt planarity. After renumbering the levels from 1 to $k$ we obtain a $k$-track planar embedding $\mathcal{E}_l$ of $G$. $\square$

This leads to our main result. Because of Lemma 2 the given transformation can be used to reduce track planarity to level planarity in linear time.

**Theorem 1.** *There is an $\mathcal{O}(|V|)$ time reduction of track planarity to level planarity.*

## 4 Algorithm

Because of Theorem 1 we obtain a straightforward track planarity testing algorithm. $G$ is transformed into $G'$ and any level planarity test can be applied afterwards. Because the transformation of $G$ runs in linear time, the overall algorithm has the same time complexity as the embedded level planarity test, $\mathcal{O}(|V|)$ for Leipert's algorithm and up to $\mathcal{O}(|V|^4)$ for Healy and Kuusik's algorithm for non-proper graphs. Note that our construction has to be made proper to be used with the $\mathcal{O}(|V|^2)$ time algorithm of Healy and Kuusik.

As Algorithm 1 shows, the procedure to obtain a track planar embedding $\mathcal{E}_l$ is nearly the same as for the test case using a level planar embedding algorithm.

**Algorithm 1:** TRACK-PLANAR-EMBED

**Input**: A track graph $G = (V, E, \phi)$
**Output**: $\mathcal{E}_l$ if $G$ is track planar, $\emptyset$ otherwise

remove all isolated vertices from $G$
$G' \leftarrow$ TRANSFORM($G$)
$\mathcal{E}_l \leftarrow$ LEVEL-PLANAR-EMBED($G'$)
**if** $\mathcal{E}_l = \emptyset$ **then return** $\emptyset$
remove all levels $i$ from $\mathcal{E}_l$ for which $i \not\equiv 2 \pmod 3$
renumber the levels from 1 to $k$
insert each removed isolated vertex $v$ of $G$ at the end of $\mathcal{E}_l[\phi(v)]$
  in arbitrary order
**return** $\mathcal{E}_l$

Again, the running time of the level embedding algorithm, $\mathcal{O}(|V|)$ time up to $\mathcal{O}(|V|^6)$ time, dominates the overall complexity. In order to prevent occurrences of isolated vertices within inner faces of diamonds we handle these in a special way. They are removed at the beginning and reinserted in arbitrary order afterwards. Placing them at the end of their ordered level in $\mathcal{E}_l$ does not violate planarity in any case.

**Corollary 1.** *The algorithm TRACK-PLANAR-EMBED returns a valid track planar embedding if and only if the input track graph $G$ is track planar.*

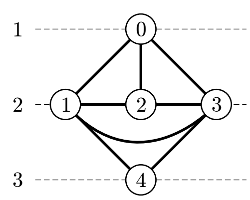
*Proof.* Assuming that the level embedding algorithm applied in Algorithm 1 returns a valid level embedding $\mathcal{E}'_l$ we obtain a valid vertex ordering of each level $i$ with $2 \equiv i \pmod 3$. These are exactly the vertex orderings of each level in a valid track planar embedding $\mathcal{E}_l$ of $G$ according to the proof of Lemma 2. □

## 5 Circle Planarity

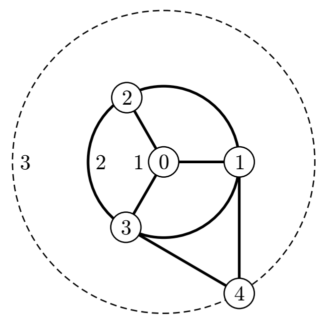
In [1,2] level planarity has been generalised to radial level planarity. In contrast to level planar graphs, the vertices of a radial planar graph are not drawn on $k$ horizontal lines but on $k$ concentric circle lines $l_i = \{\, (i\cos\theta, i\sin\theta) \mid \theta \in [0, 2\pi) \,\}$,

$1 \leq i \leq k$. A $k$-level graph is *radial $k$-level planar* or short *radial planar* if there are orderings $\leq_i$ of the vertices on each radial level such that edges are drawn as strictly monotone curves from inner to outer levels without crossings. The transformation of a level planar embedding to a radial planar embedding can be obtained by connecting the ends of each level and thus forming concentric level circles. This allows the insertion of some additional edges connecting the end of one level with the beginning of another. These *cut edges* cross an imaginary ray from the centre of the concentric levels to infinity through the points where the connected ends of the level lines meet. As an extension to level planar embeddings, *radial planar embeddings* need additional information about cut edges and their direction.

Combining radial planarity and track planarity, there is an even larger class of level graphs that can be tested for planarity efficiently. These are are called *$k$-circle planar graphs*. Circle planar graphs differ from radial planar graphs in the same way as track planar graphs differ from level planar graphs, i. e., consecutive vertices on a radial level can be connected by a "horizontal" edge. Fig. 4(b) shows a circle planar drawing of the track graph in Fig. 4(a) which is neither level planar nor track planar.




(a) A 3-track graph




(b) A circle planar drawing of Fig. 4(a)

**Fig. 4.** A circle planar graph and a circle planar drawing of it

The concept of detecting and embedding k-circle planar graphs in linear time is the same as for k-track planar graphs. At the beginning all isolated vertices are removed from the graph. The number of levels is tripled and every horizontal edge is replaced by a diamond. If in the original graph a horizontal chain closes to a *circle*, i. e., a single level cycle, we call the arising structure in the transformation a *diamond wheel*. For example, Fig. 5 is generated from the graph shown in Fig. 4. It contains a diamond wheel on levels 4 to 6. After transforming a circle graph $G = (V, E, \phi)$ into $G' = (V', E', \phi')$ the linear time radial level planarity test algorithm described in [1, 2] can be applied to $G'$ to detect circle planarity of $G$ according to the following lemma:

**Lemma 3.** *Let $G$ be a radial level graph without isolated vertices and $G'$ be the radial level graph obtained via the transformation of $G$. Then the following holds:*

$$G \text{ is } k\text{-circle planar} \Leftrightarrow G' \text{ is radial } 3k\text{-level planar}$$

*Proof.* Since $G$ does not contain isolated vertices, the proof is analogous to the proof of Lemma 2.

It remains to show what to do if a graph contains isolated vertices. If a circle occurs on level $i$, $1 \leq i \leq k$, we have no space left to place isolated vertices of $i$. Thus if this is the case the graph under test has to be rejected as non circle planar. This can be tested within linear time, e. g., in a post execution step.

A radial level embedding $E'_l$ of $G'$ can easily be transformed into a circle planar embedding $E_l$ of $G$, see Section 3. But as described in [1, 2], to compute a radial embedding the algorithm must identify cut edges. Therefore exactly one of the artificial edges $(u, v_e)$ and $(v, v_e)$ generated by the introduction of diamonds being marked as a cut edge means that its original horizontal edge $e$ is a cut edge in the radial embedding of the original graph $G$. For example, in Fig. 5 the edge $(9, 1)$ is a cut edge and therefore edge $(3, 1)$ in Fig. 4 is a cut edge, too. Edge $(4, 1)$ also is a cut edge but an original one.

**Theorem 2.** *There is an $\mathcal{O}(|V|)$ time reduction of circle planarity to radial level planarity.*

## 6 Conclusion

We presented a linear time reduction from track planarity to level planarity, leading to a linear time algorithm to recognise track planarity. It easily can be integrated into existing level planarity test algorithms. The core algorithms need not be changed because we only extend the input graph with $\mathcal{O}(|V|)$ dummy vertices and $\mathcal{O}(|V|)$ edges in a preprocessing step. Thus the algorithm time complexity does not change. Further we presented a linear time algorithm for the detection of $k$-circle planar graphs and for computing an embedding for them.

To check the practical capability of our algorithm we have realised a prototypical implementation in C++ using the technique of [22] and the Graph Template Library [12] with *improved symmetric lists* [3]. This is an efficient data structure for storing the children of a node of a PQ-tree. It is a cyclic connected list which allows reversion of its members and insertion of elements or of another (reversed) list, both in constant time.

Further investigations are desired in order to expand the test algorithms for the various kinds of level planarity for detecting the so called *minimal non level planar subgraph patterns (MNLP patterns)* if the tested graph is not level planar. These MNLP-patterns for level graphs are characterized in [13, 15] and

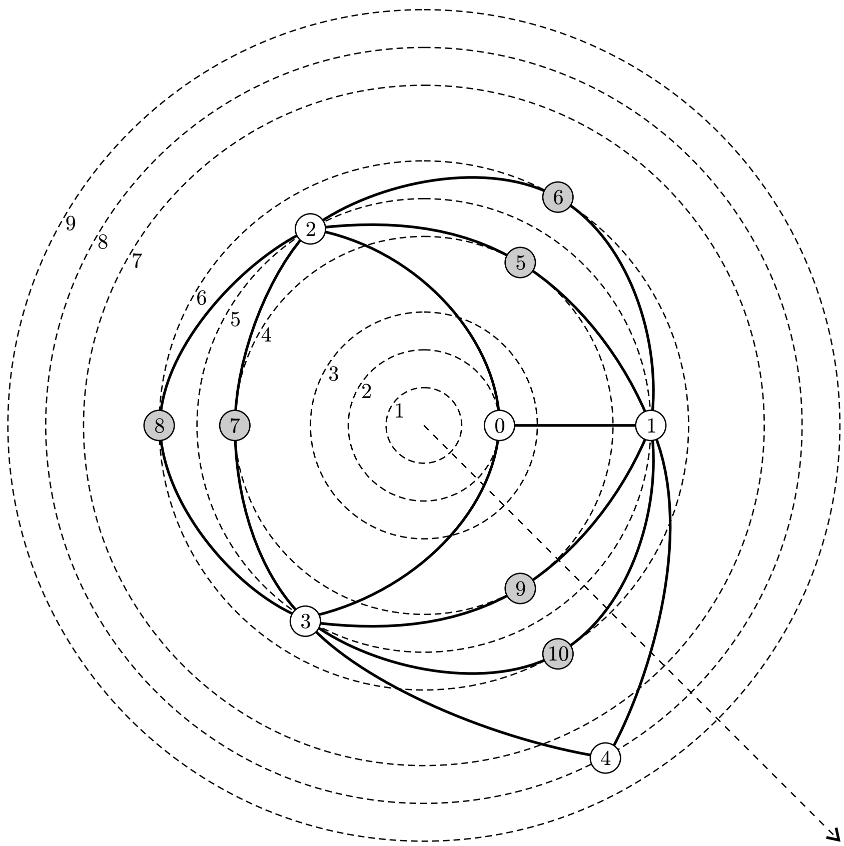



**Fig. 5.** The circle planar graph $G'$ computed from Fig. 4

are the counterparts of the Kuratowski Graphs $K_{3,3}$ and $K_5$ in planarity testing. As already mentioned in the conclusion of [22, p. 211], the detection of MNLP-patterns is especially desirable because they can also be used to verify the results of a level planarity test. Because such a test, at least the linear time algorithm, is a non trivial algorithm and thus it is not unlikely that an implementation is faulty, it is desirable not only to prove planarity with an embedding or drawing but also to show non-planarity on the basis of MNLP-patterns for debugging purposes.

## References


[1] C. Bachmaier, F. J. Brandenburg, and M. Forster. Radial level planarity testing and embedding in linear time. Technical Report MIP-0303, University of Passau, June 2003.

[2] C. Bachmaier, F. J. Brandenburg, and M. Forster. Radial level planarity testing and embedding in linear time (extended abstract). In G. Liotta, editor, *Proc. Graph Drawing, GD 2003*, LNCS. Springer, 2003. to appear.

[3] C. Bachmaier and M. Raitner. Improved symmetric lists. Submitted for publication, August 2003.

[4] K. S. Booth and G. S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *Journal of Computer and System Sciences*, 13:335–379, 1976.

[5] M. Chandramouli and A. A. Diwan. Upward numbering testing for triconnected graphs. In *Proc. Graph Drawing, GD 1995*, volume 1027 of *LNCS*, pages 140–151. Springer, 1996.

[6] H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10:41–51, 1990.

[7] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1999.

[8] G. Di Battista and E. Nardelli. Hierarchies and planarity theory. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(6):1035–1046, 1988.

[9] S. Even. *Algorithms*, chapter 7, pages 148–191. Computer Science Press, 1979.

[10] S. Felsner, G. Liotta, and S. Wismath. Straight-line drawings on restricted integer grids in two and three dimensions. In *Proc. Graph Drawing, GD 2001*, volume 2265 of *LNCS*, pages 328–342. Springer, 2002.

[11] A. Garg, M. T. Goodrich, and R. Tamassia. Planar upward tree drawings with optimal area. *International Journal of Computational Geometry and Applications*, 6(3):333–356, 1996.

[12] GTL. Graph Template Library. `http://www.infosun.fmi.uni-passau.de/GTL/`. University of Passau.

[13] P. Healy and A. Kuusik. Characterisation of level non-planar graphs by minimal patterns. Technical Report UL-CSIS-98-4, Department of Computer Science and Information Systems, University of Limerick, July 1998.

[14] P. Healy and A. Kuusik. The vertex-exchange graph: A new concept for multi-level crossing minimisation. In *Proc. Graph Drawing, GD 1999*, volume 1731 of *LNCS*, pages 205–216. Springer, 1999.

[15] P. Healy, A. Kuusik, and S. Leipert. Characterization of level non-planar graphs by minimal patterns. In D.-Z. Du, P. Eades, V. Estivill-Castro, X. Lin, and A. Sharma, editors, *Computing and Combinatorics, 6th Annual International Conference, COCOON 2000*, volume 1858 of *LNCS*, pages 74–84. Springer, 2000.

[16] L. S. Heath and S. V. Pemmaraju. Recognizing leveled-planar dags in linear time. In *Proc. Graph Drawing, GD 1995*, volume 1027 of *LNCS*, pages 300–311. Springer, 1996.

[17] L. S. Heath and S. V. Pemmaraju. Stack and queue layouts of directed acyclic graphs: Part II. *SIAM Journal on Computing*, 28(5):1588–1626, 1999.

[18] L. S. Heath and A. L. Rosenberg. Laying out graphs using queues. *SIAM Journal on Computing*, 21(5):927–958, 1992.

[19] M. Jünger and S. Leipert. Level planar embedding in linear time. In *Proc. Graph Drawing, GD 1999*, volume 1731 of *LNCS*, pages 72–81. Springer, 1999.

[20] M. Jünger and S. Leipert. Level planar embedding in linear time. *Journal of Graph Algorithms and Applications*, 6(1):67–113, 2002.

[21] M. Jünger, S. Leipert, and P. Mutzel. Level planarity testing in linear time. In *Proc. Graph Drawing, GD 1998*, volume 1547 of *LNCS*, pages 224–237. Springer, 1998.

[22] S. Leipert. *Level Planarity Testing and Embedding in Linear Time*. Dissertation, Mathematisch-Naturwissenschaftliche Fakultät der Universität zu Köln, 1998.

[23] A. Lempel, S. Even, and I. Cederbaum. An algorithm for planarity testing of graphs. In P. Rosenstiehl, editor, *Theory of Graphs, International Symposium, Rome, July 1966*, pages 215–232. Gordon and Breach, 1967.