

# Drawing Graphs on Two and Three Lines<sup>\*</sup>

Sabine Cornelsen, Thomas Schank, and Dorothea Wagner

University of Konstanz, Department of Computer & Information Science  
{Sabine.Cornelsen, Thomas.Schank, Dorothea.Wagner}@uni-konstanz.de

**Abstract.** We give a linear-time algorithm to decide whether a graph has a planar LL-drawing, i.e. a planar drawing on two parallel lines. This has previously been known only for trees. We utilize this result to obtain planar drawings on three lines for a generalization of bipartite graphs, also in linear time.

## 1 Introduction

Let  $G = (A \cup B, E)$  be a *partitioned graph*, i.e. a graph with a partition of its vertex set into two disjoint sets  $A$  and  $B$ . We will refer to the vertices of  $A$  as  $A$ -vertices and to the vertices of  $B$  as  $B$ -vertices, respectively. The question of drawing partitioned graphs on parallel lines arises from drawing bipartite graphs, i.e. partitioned graphs such that  $A$  and  $B$  are independent sets. A natural way to draw such graphs is to draw all vertices of  $A$  on one – say horizontal – line, all vertices of  $B$  on a parallel line, and all edges as straight-line segments between their end-vertices. Such a drawing will be denoted by *BA-drawing*.

If  $G$  is planar, a drawing without edge crossings would be desirable. Harary and Schwenk [6] and Eades et al. [3] showed that a bipartite graph  $G$  has a planar *BA-drawing* if and only if  $G$  is a caterpillar, i.e. a tree such that the set of all vertices of degree larger than one induces a path.

To obtain planar drawings of a larger class of bipartite graphs, Fößmeier and Kaufmann [5] proposed *BAB-drawings*. Again, every edge is a straight-line segment between its end-vertices and all vertices of  $A$  are drawn on one horizontal line, but the vertices of  $B$  may be drawn on two parallel lines – one above the  $A$ -vertices and one below. Fößmeier and Kaufmann [5] gave a linear-time algorithm to test whether a bipartite graph has a planar *BAB-drawing*. Another generalization of planar *BA-drawings* of bipartite graphs are planar drawings of leveled graphs which are considered by Jünger et al. [7].

Planar drawings for non-bipartite partitioned graphs are considered by Biedl et al. [1,2]. A complete characterization of graphs that have a planar *BA-drawing* is given in [1]. Felsner et al. [4] considered line-drawings of unpartitioned graphs. They gave a linear time algorithm that decides whether a tree has a planar straight line drawing on a fixed number of lines.

We will show how to decide in linear time whether an arbitrary graph has a planar *LL-drawing*, i.e. a straight line drawing on two parallel lines without

<sup>\*</sup> This work was partially supported by the DFG under grant BR 2158/1-1.

edge crossings. Our algorithm works even if the end-vertices of some edges are constrained to be on different lines. For planar  $BAB$ -drawings, we relax the condition of bipartiteness to partitioned graphs with the only constraint that the neighbor of each vertex of  $B$  with degree one is in  $A$ . Actually, even this restriction is only necessary in very special cases, which are discussed in Section 2.2. Note that LL-drawings of unpartitioned graphs are the special case of  $BAB$ -drawings with  $A = \emptyset$ .

This paper is organized as follows. In Section 2, we consider planar  $BAB$ -drawings. First, we decompose the input graph such that the  $A$ -vertices of each component induce a path. We then show that we can substitute each of these components by a graph that contains only  $B$ -vertices, but simulates the possible planar  $BAB$ -drawings of the component. Finally, in Section 3, we show how to decide whether an unpartitioned graph has a planar drawing on two parallel lines.

## 2 $BAB$ -Drawings

Let  $G = (A \cup B, E)$  be a partitioned graph such that every  $B$ -vertex of degree one is adjacent to an  $A$ -vertex. Since we want to test planarity, we can reject every graph with more than  $3n - 6$  edges. So we can assume that the number of edges is linear in the number  $n := |A \cup B|$  of vertices. For a subset  $S \subset A \cup B$  we denote by  $G(S)$  the graph that is induced by  $S$ .

### 2.1 Decomposition

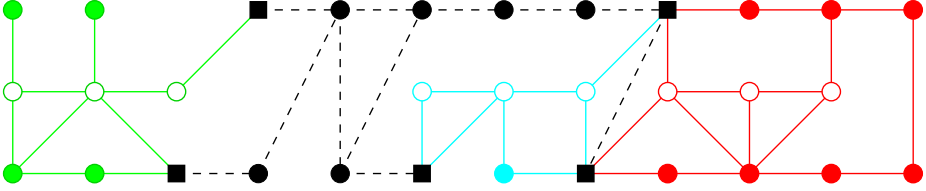
If  $G$  has a planar  $BAB$ -drawing, the connected components of  $G(A)$  have to be paths. Therefore, the vertex set of a connected component of  $G(A)$  will be called an  $A$ -path of  $G$ . By  $\mathcal{P}(A)$ , we denote the set of  $A$ -paths.  $\mathcal{P}(A)$  can be determined in linear time. Next, we want to decompose  $G$  into components. A *subdivision  $B$ -path between two vertices  $b_1$  and  $b_k$*  is a set  $\{b_2, \dots, b_{k-1}\} \subset B$  such that

- $\{b_i, b_{i+1}\} \in E$  for  $i = 1 \dots k - 1$  and
- degree  $b_i = 2$  for  $i = 2 \dots k - 1$ .

For an  $A$ -path  $P \in \mathcal{P}(A)$ , the  $A$ -path component  $G_P$  is the graph induced by the union of the following sets of vertices

- set  $P$ ,
- set  $B_P$  of all  $B$ -vertices that are incident to  $P$ ,
- all *subdivision  $B$ -paths* between two vertices of  $B_P$  that are not *subdivision  $B$ -paths* between two vertices of  $B_{P'}$  for any  $P' \in \mathcal{P}(A) \setminus \{P\}$ .

Similar to the subdivision paths, we also omit edges that would be contained in several  $A$ -path components in any of these components. Figure 1 shows a graph with three  $A$ -path components. A vertex of an  $A$ -path component  $G_P$  that is adjacent to a vertex of  $G - G_P$  is called a *connection vertex* of  $G_P$ . Given a  $BAB$ -drawing of  $G$ , we call the first and last vertex of  $G_P$  on each of the



**Fig. 1.** Decomposition of a graph into three  $A$ -path components. White vertices are  $A$ -vertices, all other vertices are  $B$ -vertices. Rectangularly shaped vertices are connection vertices. Dashed edges are not contained in any of the three  $A$ -path components.

three lines a *terminal* of  $G_P$ . By the restriction on  $B$ -vertices of degree one, the following lemma is immediate.

**Lemma 1.** *Let  $P$  be an  $A$ -path of  $G$ .*

1. *All connection vertices of  $G_P$  are in  $B_P$ .*
2. *In any planar  $BAB$ -drawing of  $G$ , every connection vertex of  $G_P$  is a terminal of  $G_P$ .*
3. *If  $G$  has a planar  $BAB$ -drawing,  $G_P$  has at most four connection vertices.*

Let  $P, P' \in \mathcal{P}(A)$  be distinct and  $b_1, b_2 \in B_P \cap B_{P'}$  be two connection vertices of both  $G_P$  and  $G_{P'}$ . Then  $b_1$  and  $b_2$  are drawn on different lines in any planar  $BAB$ -drawing. In this case, we add edge  $\{b_1, b_2\}$  to  $G$ . We will refer to such an edge as a *reminder edge*.

- Lemma 2.**
1. *The sum of the sizes of all  $A$ -path components is in  $\mathcal{O}(n)$ .*
  2. *There is a linear-time algorithm that either computes all  $A$ -path components and reminder edges or returns an  $A$ -path component that has more than four connection vertices.*

*Proof.* 1. By definition, each edge is contained in at most one  $A$ -path component. The number of  $A$ -path components is at most  $|A|$  and each  $A$ -path component is connected. Thus the sum of the number of vertices in all  $A$ -path components is at most  $|E| + |A| \in \mathcal{O}(n)$ .

2. First, each subdivision  $B$ -path is substituted by a single edge between its two end-vertices and all sets  $B_P$  are computed. This can be done in linear time, e.g. by depth first search. Then for each  $P \in \mathcal{P}(A)$ , we examine the incident edges  $e$  of all  $b \in B_P$ . If both end vertices of  $e$  are contained in  $B_P \cup P$ , we add  $e$  to  $G_P$ , if not, we classify  $b$  as a connection vertex and stop the examination of this vertex  $b$ . This guarantees that for each vertex, at most one edge that is not in  $G_P$  is touched. If the number of connection vertices of  $G_P$  is greater than 4, we can return  $G_P$ , else we add a ‘pre-reminder’ edge labeled  $G_P$  between all six pairs of connection vertices of  $G_P$  to  $G$ .

In a final walk through the adjacency list of each connection vertex, we can use the pre-reminder edges to determine the reminder edges and those edges of  $G$  between connection vertices that have to be added to an  $A$ -path component. Finally, subdivision vertices are reinserted into the replacement edges. □

## 2.2 Substitution

In this section, we show how to substitute an  $A$ -path component  $G_P$  by a graph that contains only  $B$ -vertices, but simulates all possible planar  $BAB$ -drawings of  $G_P$  in which the connection vertices are terminals. We say that two terminals of  $G_P$  are on the *same side* of  $G_P$  in a  $BAB$ -drawing if they are both first or both last vertices on their lines. They are on *different sides* if one is a first and the other one a last vertex. Essentially, there are six questions of interest:

$\#$ : How many connection vertices are contained in  $G_P$ ?

$\eta$ : Does the number of connection-vertices equal the number of  $B$ -vertices of  $G_P$ ?

$\tau$ : Does  $G_P$  have a planar  $BA$ -drawing?

$\tau_v$ : For each connection vertex  $v$  of  $G_P$ , is there a planar  $BAB$ -drawing of  $G_P$  that induces a  $BA$ -drawing of  $G_P - v$ ?

$\sigma_{vw}$ : For each pair  $v, w$  of connection-vertices of  $G_P$ , is there a planar  $BAB$ -drawing of  $G_P$ , such that  $v$  and  $w$  are on the same side of  $G_P$ ?

$\delta_{vw}$ : For each pair  $v, w$  of connection-vertices of  $G_P$ , is there a planar  $BAB$ -drawing of  $G_P$ , such that  $v$  and  $w$  are on different sides of  $G_P$ ?

Note that  $\tau$  implies  $\tau_v$  and  $\tau_v$  implies  $\sigma_{vw}$  and  $\delta_{vw}$  for any pair  $v, w$  of connection vertices of  $G_P$ . Thus, provided that there exists some planar  $BAB$ -drawing of  $G_P$ , these six questions lead to the cases listed in Table 1. Note that the listed substitutes contain different kind of edges. There is one case with a parallel edge and one case with a dotted edge. Let  $G'$  be the graph constructed from  $G$  by substituting each  $A$ -path component in the way described above. We say that an edge in an  $LL$ -drawing is *vertical* if the end vertices of  $e$  are drawn on different lines.

**Lemma 3.** 1. *The size of  $G'$  is linear in the size of  $G$ .*

2.  *$G'$  has a planar  $LL$ -drawing with every multiple edge and every remainder edge drawn vertically and no dotted edge having a parallel edge if and only if  $G$  has a planar  $BAB$ -drawing.*

We discuss only, where the precondition on degree one vertices is needed and omit the rest of the proof. If we have a planar  $BAB$ -drawing and we want to construct a planar  $LL$ -drawing of  $G'$  by replacing each  $A$ -path component  $G_P$  by its substitute, we have to make sure, that for any two vertices  $v$  and  $w$  of  $G_P$  all vertices that are drawn between  $v$  and  $w$  do also belong to  $G_P$ . But there might be also subdivision paths  $S$  between vertices of degree one and vertices  $b \in B_P$ . If  $b$  is not a connection vertex, we can add  $S$  to  $G_P$ , but if  $b$  is a connection vertex, there need not exist a suitable substitute any more. There is no problem, if there are at least three subdivision paths between  $b$  and vertices of degree one, since then at least one of them has to be drawn on the other line then  $b$  and then any of them can be drawn there, thus neither has to be added to  $G_P$ .

**Lemma 4.** *For each component, the six questions that determine the substitute, can be answered in time linear in the size of the component.*

**Table 1.** Substitutes for the  $A$ -path components. Cases that correspond up to the names of the connection vertices are omitted. A + means that the corresponding question is answered by yes. A  $\oplus$  means that the answer to this property is implied. No entry means that the according property is not fulfilled.

1 connection vertex			
$\eta$	$\tau$	$\tau_v$	
+	$\oplus$	$\oplus$	$v$
	+	$\oplus$	$v - \bullet$
		+	$v$   \ / $\bullet - \bullet$
			$v - \bullet$   /   $\bullet - \bullet$

4 connection vertices			
$\sigma_{vx}$	$\sigma_{vy}$	$\sigma_{vz}$	
+	+	+	$v - \bullet$ $\bullet - y$   \ / $w - \bullet - \bullet - x$
+	+		$v - \bullet - \bullet - x$        $y - \bullet - \bullet - z$
	+		$v - \bullet - x$        $y - \bullet - z$

2 connection vertices						
$\eta$	$\tau$	$\tau_v$	$\tau_w$	$\sigma_{vw}$	$\delta_{vw}$	
+	+	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$v - w$
+		$\oplus$	$\oplus$	$\oplus$	$\oplus$	$v$    $w$
	+	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$v - \bullet - w$
		+	+	$\oplus$	$\oplus$	$v - w$   / $\bullet$
		+		$\oplus$	$\oplus$	$v - \bullet$   \ /   $w - \bullet$
				+	+	$v - \bullet - \bullet$        $w - \bullet - \bullet$
				+		$v - \bullet - \bullet$             $w - \bullet - \bullet$
					+	$v - \bullet - \bullet$             $\bullet - \bullet - w$

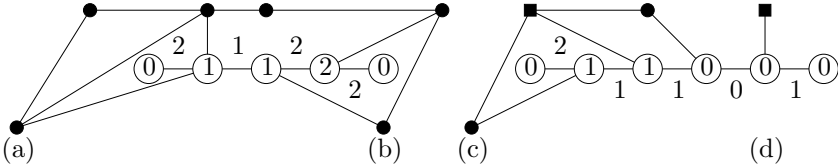
3 connection vertices						
$\tau_v$	$\tau_w$	$\tau_x$	$\sigma_{vw}$	$\sigma_{vx}$	$\sigma_{wx}$	
+	+	+	$\oplus$	$\oplus$	$\oplus$	$v - w$   / $x$
+	+		$\oplus$	$\oplus$	$\oplus$	$v - \bullet - x$   / $w$
		+	+	$\oplus$	$\oplus$	$v - \bullet - \bullet - w$   / $x$
		+		$\oplus$	$\oplus$	$x - v$   \ /   $w - \bullet$
			+	+	+	$v - \bullet$ $\bullet - \bullet$   \ / $w - \bullet - \bullet - \bullet - x$
				+	+	$v - \bullet$   \ /   $x - w$
			+			$v - \bullet - \bullet$             $w - \bullet - x$

*Proof.* It is obvious how to decide Properties # and  $\eta$ . To decide properties  $\sigma_{vw}$  and  $\delta_{vw}$ , suppose, we have an  $A$ -path component  $G_P$  with an  $A$ -path  $P : a_1, \dots, a_k$ . For technical reasons, we add vertices  $a_0$  and  $a_{k+1}$ , one at each end, to  $P$ .

Let  $W$  be the set of vertices of a connected component of  $G_P(B)$ . Suppose first that we wouldn't allow edges between  $B$ -vertices on different lines. In this case  $G(W)$  has to be a path and the vertices in  $W$  have to occur in the same order as their adjacent vertices in  $P$ . Furthermore, let  $a_\ell$  ( $a_r$ ) be the  $A$ -vertex

with lowest (highest) index that is adjacent to  $W$ . Then we say that the edges  $\{a_i, a_{i+1}\}, i = \ell, \dots, r - 1$  are *occupied* by  $G(W)$ . Finally, if a vertex  $b$  in  $G(W)$  is a connection vertex of  $G_P$  then  $b$  has to be a terminal. Thus, if  $a_\ell$  is an  $A$ -vertex with the lowest (highest) index that is adjacent to  $b$ , then the edges  $\{a_i, a_{i+1}\}, i = 0, \dots, \ell$  (or  $i = \ell, \dots, k + 1$ ) are also occupied by  $G(W)$ .

In general, if  $G_P$  has a planar  $BAB$ -drawing,  $G(W)$  might be a path with at most two chords and the  $A$ -vertices adjacent to this path may change at most twice between increasing and decreasing order. Thus, there are three reasons for changing the line in  $W$ : a chord, a change in the order of the adjacent  $A$ -vertices, or a connection vertex. Note also that such a change of lines might occur in at most two connected components of  $G_P(B)$ . If there is a line change between  $b_1$  and  $b_2$  in  $W$ , then similar to the case of connection vertices above,  $G(W)$  also occupies the edges between the adjacent  $A$ -vertices of  $b_1$  and  $b_2$  and one end of the  $A$ -path  $P$ . Note that in case of a line change some edges in  $G(P)$  might be occupied twice.



**Fig. 2.** Three reasons for changing a line: a) a chord, b) a change in the order of adjacent  $A$ -vertices, and c) a connection vertex. d) A connection vertex occupies a part of the  $A$ -path. The numbers indicate how many times an edge or a vertex in the  $A$ -path is occupied.

We label every edge and every vertex in  $G(P)$  with the number of times it is occupied as indicated in Fig. 2, where the number of times a vertex  $v \in P$  is *occupied* by  $G(W)$  is defined as follows.  $v$  is occupied  $k$  times by  $G(W)$ , if both adjacent edges in the  $A$ -path are occupied  $k$ -times by  $W$  and  $v$  is not adjacent to an end vertex of  $W$  and  $k - 1$  times if it is adjacent to such an end vertex. It is also occupied  $k - 1$ -times if one adjacent edge is occupied  $k$ -times and the other one  $k - 1$  times. There exists a planar  $BAB$ -drawing with the corresponding choice of the direction of the connection vertices, if and only if

1. in total, every edge in  $G(P)$  is occupied at most twice and
2. each vertex  $v \in P$  that is adjacent to an end vertex of a connected component of  $G_P(B)$  is occupied at most once.

This test works in linear time, since we can reject the graph, as soon as an edge gets a label higher than 2 and because there are at most four connection vertices and at most two components that change lines for which different orientations have to be checked. A similar approach can be used to answer questions  $\tau$  and  $\tau_v$ .  $\square$

The results in the next section show how to test the conditions of Lemma 3,2, which completes the algorithm for deciding whether a planar *BAB*-drawing exists.

### 3 LL-Drawings

Given a graph  $G = (V, E)$  and a set of edges  $U \subset E$ , we show how to decide in linear time whether there exists a planar LL-drawing of  $G$  with the property that each edge in  $U$  is drawn vertically. A long version of the planar LL-drawability test can be found in [8]. We will first discuss trees and then the general case.

#### 3.1 LL-Drawings of Trees

Felsner et al. [4] characterized trees that have a planar LL-drawing and called them strip-drawable. We give a slightly different characterization.

**Lemma 5.** *A tree  $T$  has a planar LL-drawing, if and only if it contains a spine, i.e. a path  $S$  such that  $T - S$  is a collection of paths.*

*Proof.*  $\Rightarrow$  The unique path between a leftmost and a rightmost vertex of  $T$  in a planar LL-drawing is a spine of  $T$ .

$\Leftarrow$  An LL-drawing can be achieved by placing the spine on one of the lines and the path components in the order of their adjacency to the spine on the other line. □

As indicated in [4], the inclusion minimal spine  $S_{\min}$  can be computed in linear time. If a vertex  $v$  is required to be an end vertex of a spine, we only have to check if there is a path between  $v$  and an end vertex of  $S_{\min}$  in  $T - S_{\min}$ . We will use this fact when we examine the general case. Finally, the following lemma characterizes whether the edges in a set  $U \subset E$  can be drawn vertically.

**Lemma 6.** *An edge  $e$  can be drawn vertically if and only if there exists a spine  $S$  that contains at least one end vertex of  $e$ .*

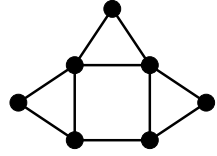
#### 3.2 LL-Drawings of General Graphs

For general graphs, the existence of a spine is still necessary for the existence of a planar LL-drawing, but it is not necessarily sufficient. For example, see the



**Fig. 3.** LL-drawings of a tree. The spine is drawn black.

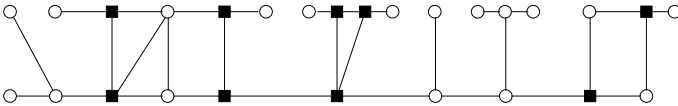
graph on the right. To test whether a general graph has a planar LL-drawing, we first split the graph into components like paths, trees and two connected components. We give necessary conditions for these components and show how to test them in linear time. Finally, by constructing a drawing, we show that these necessary conditions are also sufficient. First we make the following observation.



**Lemma 7.** *Every graph that has a planar LL-drawing is outer planar.*

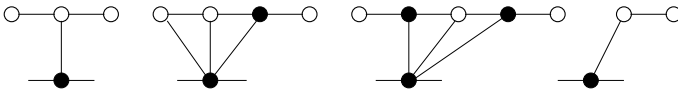
*Proof.* Since all vertices are placed on two horizontal lines, every vertex is either a top most or bottom most vertex and is thus incident to the outer face.  $\square$

Suppose now without loss of generality that  $G$  is a connected graph. A vertex  $v$  is called a *connection vertex*, if it is contained in a cycle and its removal disconnects  $G$ . We denote the set of all connection vertices by  $V_c$ . The connection vertices can be determined with depth first search in linear time.



**Fig. 4.** A graph with a planar LL-drawing. Connection vertices are solid.

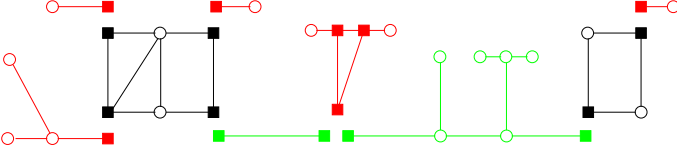
A subgraph  $L$  of  $G$  is called a *single line component*, if it is maximal with the property that  $L$  is an induced graph and there exists a vertex  $v \in V_c$  such that  $L$  is a connected component of  $G - v$ . By  $\bar{L}$ , we denote a single line component  $L$  including its incident connection vertex. If  $\bar{L}$  forms a path we call it a *strict single line component*, otherwise we call it a *fan*.



**Fig. 5.** Fans and a strict single line component, connection vertices are solid.

Simply testing for all  $v \in V_c$ , whether the connected components of  $G(V \setminus \{v\})$  are paths leads to a quadratic time algorithm for finding the single line components. But we can use outer-planarity to do it in linear time. Note that any single line component contains at most two connection vertices (see Fig. 5 for illustration). Thus, we can find them, by storing the three last visited connection vertices on a walk around the outer face and testing, whether the vertices between the first and second occurrence of the same connection vertex induce a path.

Let  $G'$  be the subgraph that results from  $G$  by deleting all single line components. The *two-lined components* of  $G$  are the components that we get by splitting  $G'$  at the connection vertices of  $G$ . Two-lined components that do not contain a cycle will be called *tree components*. All other two-lined components will be denoted by *two-lined biconnected components*. Figure 6 illustrates the different components of the graph in Fig. 4.



**Fig. 6.** Components of the graph in Fig. 4. There are two trees, two two-lined biconnected components, two fans, and three strictly single lined components.

We have now defined all components that we need to characterize those graphs that have a planar LL-drawing. Note that a graph that contains no two-lined biconnected components has a planar LL-drawing if and only if it contains a spine. Before we formulate the characterization for the general case, consider the set  $\mathcal{L}$  of all two-lined components. Let  $\mathcal{P}$  be the graph with vertex set  $\mathcal{L}$  in which two two-lined components are adjacent if and only if they share a connection vertex. Suppose a planar LL-drawing for  $G$  is given. Then all two-lined biconnected components require at least two lines. All tree components also require at least two lines or are connected to two components that require two lines. Thus  $\mathcal{P}$  has to be a path. We will refer to this property by saying that *the two-lined components induce a path*.

**Theorem 1.** *A graph  $G$  has a planar LL-drawing if and only if*

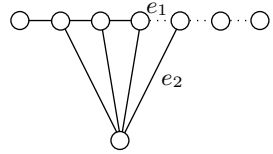
1.  $G$  is outer planar,
2. the two-lined components induce a path,
3. each tree component  $T$  has a spine  $S$  such that the connection vertices of  $T$  are end vertices of  $S$ , and
4. for each two-lined biconnected component  $B$  there is a drawing with the following properties.
  - (a) Connection vertices are leftmost or rightmost vertices of  $B$  on their line.
  - (b) At most one two-sided component is connected to each vertex of  $B$ .
  - (c) If a two-sided component or a fan is connected to a vertex of  $B$ , then a vertex on the same side is at most connected to a path.

The necessity of these conditions follows from Section 3.1, Lemma 7, and the observations mentioned above. To show that these conditions are also sufficient, we sketch how to find a drawing of  $G$  in linear time.

**Drawing and Sufficient Conditions.** In this subsection we sketch how to construct in linear time a planar LL-drawing if the conditions in Theorem 1 are fulfilled. We also mention which edges can be drawn vertically. Since a linear order on the two-lined components is given, we only have to show how to draw each component separately.

It was already shown in Section 3.1 how to find a spine and that a tree with a spine has a planar LL-drawing. For drawing two-lined biconnected components, first note that a biconnected graph has a unique outer planar embedding. Starting with a connection vertex on any line, we add the others in the order of their appearance around the outer face and switch lines only if necessary, i.e. at a connection vertex or a chord. We do this procedure for each direction around the outer face. Those cases in which lines are switched at most twice, are the possible outer planar drawings of a two-lined biconnected component, with respect to Theorem 1.4. Hence, if a drawing exists, it can be found in linear time.

Vertical edges in two-lined biconnected components only yield another condition for switching lines. Let  $\bar{L}$  be a fan on a single line component  $L$  and a connection vertex  $v$ . If  $\bar{L}$  is connected to two two-lined components, only edges between  $v$  and  $L$  can be vertical edges. Else, also edges indicated by dotted lines in the Figure on the right hand side can be drawn upright, but only one of the edges  $e_1$  and  $e_2$  simultaneously.



## 4 Conclusion

We showed how to decide in linear time, whether an arbitrary graph has a planar drawing on two parallel lines, even if the input contains edges that have to be drawn vertically. We applied this result to decide in linear time whether a partitioned graph  $G = (A \cup B, E)$  with the property that every  $B$ -vertex of degree one is adjacent to an  $A$ -vertex has a planar  $BAB$ -drawing. The algorithm worked in three steps. First, the graph was decomposed into  $A$ -path components. Then, each of these components was substituted by a graph that contained only  $B$ -vertices, but simulates the possible positions of the connection vertices. Finally, we test whether the resulting graph has a planar LL-drawing. We discussed that the restriction on the vertices of degree one is only needed in the following sense: If  $b$  is a connection vertex of an  $A$ -path component, the number of subdivision  $B$ -paths between  $b$  and vertices of degree one may not be one or two.

For further research, it would be interesting to examine whether our approach could be extended to  $BAAB$ -drawings, that is drawings for which there are two parallel middle lines for the  $A$ -vertices.

## References

1. T. C. Biedl. Drawing planar partitions I: LL-drawings and LH-drawings. In *Proceedings of the 14th Annual ACM Symposium on Computational Geometry (SCG '98)*, pages 287–296, 1998.

2. T. C. Biedl, M. Kaufmann, and P. Mutzel. Drawing planar partitions II: HH-drawings. In J. Hromkovic, editor, *Graph Theoretic Concepts in Computer Science, 24th International Workshop, (WG '98)*, volume 1517 of *Lecture Notes in Computer Science*, pages 124–136. Springer, 1998.
3. P. Eades, B. D. McKay, and N. C. Wormald. On an edge crossing problem. In *Proceedings of the 9th Australian Computer Science Conference (ACSC 9)*, pages 327–334, 1986.
4. S. Felsner, G. Liotta, and S. K. Wismath. Straight-line drawings on restricted integer grids in two and three dimensions. In M. Jünger and P. Mutzel, editors, *Proceedings of the 9th International Symposium on Graph Drawing (GD 2001)*, volume 2265 of *Lecture Notes in Computer Science*, pages 328–342. Springer, 2002.
5. U. Fößmeier and M. Kaufmann. Nice drawings for planar bipartite graphs. In G. Bongiovanni, D. P. Bovet, and G. Di Battista, editors, *Proceedings of the 3rd Italian Conference on Algorithms and Complexity (CIAC '97)*, volume 1203 of *Lecture Notes in Computer Science*, pages 122–134. Springer, 1997.
6. F. Harary and A. Schwenk. A new crossing number for bipartite graphs. *Utilitas Mathematica*, 1:203–209, 1972.
7. M. Jünger, S. Leipert, and P. Mutzel. Level planarity testing in linear time. In S. H. 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.
8. T. Schank. Algorithmen zur Visualisierung planarer partitionierter Graphen. Master's thesis, Universität Konstanz, 2001. <http://www.inf.uni-konstanz.de/algo/lehre/theses/>.