Track Drawings of Graphs with Constant Queue Number*

(Technical Report 2003-474, School of Computing, Queen's University)

Emilio Di Giacomo †   *Henk Meijer*‡

digiacomo@diei.unipg.it   henk@cs.queensu.ca

Abstract

A k-track drawing is a crossing-free $3D$ straight-line drawing of a graph G on a set of k parallel lines called *tracks*. The minimum value of k for which G admits a k-track drawing is called the *track number* of G. In [18] it is proved that every graph from a proper minor closed family has constant track number if and only if it has constant queue number. In this paper we study the track number of well-known families of graphs with small queue number. For these families we show upper bounds and lower bounds on the track number that significantly improve previous results in the literature. Linear time algorithms that compute track drawings of these graphs are also presented and their volume complexity is discussed.

1 Introduction and Overview

The problem of computing a drawing of a graph with small area/volume has received a lot of attention in the graph drawing literature during the last decade. However, while research devoted to computing small-sized drawings in the plane has flourished during the last two decades and has produced a rich body of combinatorial results, structures, algorithmic techniques, and software systems, the research on 3D drawings can still be considered in its early stages [5, 16].

Cohen, Eades, Lin and Ruskey [3] showed that every graph admits crossing-free 3D drawing on an integer grid of $O(n^3)$ volume, and proved that this is asymptotically optimal. The volume of a drawing is measured as the number of grid-points contained in the smallest axis-aligned box bounding the drawing. Calamoneri and Sterbini [1] showed that all 2-, 3-, and 4-colourable graphs can be drawn in a 3D grid of $O(n^2)$ volume with $O(n)$ aspect ratio and proved a lower bound of $\Omega(n^{1.5})$ on the volume of such graphs. For r-colourable graphs, Pach, Thiele and Tóth [17] showed a bound of $\theta(n^2)$ on the volume. Garg, Tamassia, and Vocca [12] showed that all 4-colorable graphs (and hence all planar graphs) can be drawn in $O(n^{1.5})$ volume and with $O(1)$ aspect ratio but by using a grid model where the coordinates of the vertices may not be integer. Chrobak, Goodrich, and Tamassia [2] gave an algorithm for constructing 3D convex drawings of triconnected planar graphs with $O(n)$ volume and non-integer coordinates.

Recent papers [6, 7, 8, 9, 10, 18] have considered the following problem: given a grid ϕ such that ϕ is a proper subset of the integer 3D space, which graphs admit a straight line crossing-free drawing with vertices located at the grid points of ϕ? If ϕ is chosen so that it has volume V, then a volume bound of V can be determined for any class of graph drawable on ϕ. A grid ϕ of this type is also called *restricted integer grid*.

Felsner et al. [10] initiated the study of restricted integer grids consisting of parallel grid lines, called *tracks*. In particular, they focus on the *box* and the 3-*prism*. A box is a grid consisting of four parallel lines, one grid unit apart from each other and a 3-prism uses three non-coplanar parallel lines. It is shown that all outerplanar graphs can be drawn on a 3-prism where the length of the lines is $O(n)$. This result gives the first algorithm to compute a crossing-free straight-line 3D drawing with linear volume for a non-trivial family of planar graphs. Moreover it is shown that there exist planar graphs that cannot be drawn on the prism and that even a box does not support all planar graphs.

Dujmovic et al. [7] show that if a graph G admits a drawing Γ on a a grid ϕ consisting of a constant number of parallel lines, then G has a linear volume upper bound. This result suggests that the focus of the research should be on

*Research partially supported by "Progetto ALINWEB: Algoritmica per Internet e per il Web", MIUR Programmi di Ricerca Scientifica di Rilevante Interesse Nazionale.

†Dipartimento di Ingegneria Elettronica e dell'Informazione, Università degli Studi di Perugia, Perugia, Italy.

‡Department of Computing and Information Science, Queen's University, Kingston, Ontario, Canada.

minimizing the number of tracks in a restricted integer grid, independent of the length of the tracks themselves. The *track number* $tn(G)$ of a graph G is the minimum number of tracks that is required to compute such a drawing.

Wood [18] shows a relationship between $tn(G)$ and another well-studied graph parameter, the *queue number* $qn(G)$ (i.e. the minimum number of queues in a queue layout of G [15]). He proves that every graph from a proper minor closed family has constant track number if and only if it has constant queue number. By the result of Wood all families of graphs whose queue number is known to be constant (for example series-parallel graphs, Halin graphs, Benes networks, arched leveled planar graphs, X-trees, unicyclic graphs), have a three dimensional straight-line grid drawing with linear volume. A recent result by Dujmović and Wood [8, 9] shows that linear volume can also be achieved for graphs with bounded tree width. We observe however that value of the track number (and hence of the volume) deriving from the results in [18] are often very large. In [18] it is shown that $tn(G) \leq c(2(c-1)qn(G)+1)^{c-1}$ where c is the star chromatic number of G ($c \geq 3$ for any graph). Therefore, even for the (apparently innocent) family of planar graphs whose queue number is 1 we obtain an upper bound for the track number of at least 75 and by using the technique in [7] a drawing with a bound on the volume of $75 \times 151 \times 151 \lceil \frac{n}{75} \rceil$. From the observation above it is natural to ask whether it is possible to reduce the bounds on track number and volume for graphs with constant queue number.

In this paper we present new lower and upper bounds on the track number (and hence on the volume) of some families of graphs that are known to have constant queue number. Our main contributions can be listed as follows.

- The family of graphs with queue number one (i.e. arched leveled planar graphs) are proved to have track number at least four and at most five. A drawing algorithm is presented for arched leveled planar graphs that gives a volume bound of $3 \times 3 \times n$.
- The track number of X-trees (that have queue number 2) is proved to be three and a volume bound of $2 \times 2 \times \frac{4(n+1)}{7}$ is shown.
- A lower bound of three and an upper bound of four is presented for Halin Graphs. A volume bound of $2 \times 2 \times n$ is also shown.

The drawing algorithm described in order to prove the above results all have $O(n)$ time complexity and use integer arithmetic.

Table 1 shows the upper and lower bounds on track number of the families of graphs studied in this paper and compare them with the existing ones. Table 2 shows the upper bounds on volume for the families of graphs studied in this paper and compare them with the existing ones.

Family	**Queue Number**		**Track Number**		**Previous Track Number**	
	Ω	O	Ω	O	Ω	O
Arched Leveled Planar	1	1	4	5	3	$c(2(c-1)q+1)^{c-1} \geq 75$
Meshes	1	1	3	3	3	$c(2(c-1)q+1)^{c-1} \geq 75$
X-trees	2	2	3	3	3	$c(2(c-1)q+1)^{c-1} \geq 243$
Halin	2	3	4	5	3	$c(2(c-1)q+1)^{c-1} \geq 507$
c is the star chromatic number of the graph ($c \geq 3$).						

Table 1: Upper and lower bounds on queue number and track number for different classes of graphs.

Family	**Volume**	**Previous Volume**
Arched Leveled Planar	$3 \times 3 \times n$	$2t \times p_{2t} \times p_{2t} \lceil \frac{n}{t} \rceil \geq 150 \times 151 \times 151 \lceil \frac{n}{75} \rceil$
Meshes	$2 \times 2 \times \frac{(n+2\sqrt{n})}{3}$	$2t \times p_{2t} \times p_{2t} \lceil \frac{n}{t} \rceil \geq 150 \times 151 \times 151 \lceil \frac{n}{75} \rceil$
X-trees	$2 \times 2 \times \frac{4(n+1)}{7}$	$2t \times p_{2t} \times p_{2t} \lceil \frac{n}{t} \rceil \geq 486 \times 487 \times 487 \lceil \frac{n}{243} \rceil$
Halin	$2 \times 2 \times n$	$2t \times p_{2t} \times p_{2t} \lceil \frac{n}{t} \rceil \geq 1014 \times 1019 \times 1019 \lceil \frac{n}{507} \rceil$
t is the track number of the graph; p_{2t} is the smallest prime greater than $2t$.		

Table 2: Upper bounds on the volume of a straight-line integer grid drawing for different classes of graphs.

The remainder of of the paper is organized as follows. In Section 2 some definitions and results about track and queue layout are recalled. The bounds on the track number and on the volume of arched leveled planar graphs are proved in Section 3. Results on X-trees are presented in Section 4. The track number of Halin Graphs is studied in Section 5. For reasons of space some detail are omitted and can be found in the Appendix.

2 Preliminaries

In this section we give some preliminary definitions that will be used throughout the paper and recall some known results about track layout.

2.1 Preliminary Definitions

Let $G = (V, E)$ be a graph. A *track assignment* [8, 9] of G consists of a partition $\{t_i \mid i \in I \subseteq \mathbb{N}\}$ of V, and of a total ordering $<_i$ of the vertices in each set t_i. Each set t_i is called a *track*. An *overlap* in a track assignment consists of three vertices u, v, and w such that they are in the same track t_i, there exists the edge (u,w) and $u <_i v <_i w$. An *X-crossing* in a track assignment consists of two edges (u_0,v_0) and (u_1,v_1) such that u_0 and u_1 are on the same track t_i, v_0 and v_1 are on another track t_j $(i \neq j)$ with $u_0 <_i u_1$ and $v_1 <_j v_0$. Figure 1(b) shows an example of track assignment for the graph in Figure 1(a). Vertices v_1, v_5 and v_2 form an overlap, as well as vertices v_3, v_6 and v_4. Edges (v_5,v_4), (v_2,v_3) form an X-crossing. Another X-crossing is formed by edges (v_6,v_8) and (v_4,v_7).

A *track layout* [8, 9] is a track assignment with no overlaps and no X-crossings. Figure 1(c) shows an example of a track layout of the graph of Figure 1(a). A track layout with k tracks is also called a k-track layout. The *track number* of a graph G, denoted by $tn(G)$, is the minimum k such that G has a k-track layout. A set of k tracks is also called a *k-prism*.

In the rest of the paper a track layout will be specified by assigning to each vertex v two numbers: $track(v)$ is an integer that denotes the track to with v is assigned; $order(v)$ is an integer that denotes the ordering of v in $track(v)$. We say that $u <_i v$ if $track(u) = track(v) = i$ and $order(u) < order(v)$. We shall sometimes simplify the notation and write $u < v$ instead of $u <_i v$.

A *track line* is a straight line of a 3D grid parallel to the x-axis. A *strip* σ_{ij} is the portion of a plane delimited by track lines i and j. We denote as (x, y_p, z_p) a track line passing through point (x_p, y_p, z_p). A *track drawing* of a graph G on k track lines is a 3D straight-line crossing-free grid drawing of G such that each vertex of G is drawn on one of k track lines. A track drawing on k track lines is also called k-track drawing. The drawing in Figure 1(d) is a track drawing of the graph of Figure 1(a).

In [7] a general technique is described for computing a track drawing of a graph G from a track layout of G. In this paper we will describe ad-hoc techniques that results in a smaller upper bounds on the volume. In this case we need to prove that there is no crossing between edges. Let $e_0 = (u_0,v_0)$ and $e_1 = (u_1,v_1)$ be two edges of a graph G. If two edges e_0 and e_1 cross each other in a drawing of G, then the four points representing u_0, v_0, u_1 and v_1 are co-planar, i.e. the following equation is satisfied:

$$\begin{vmatrix} 1 & 1 & 1 & 1 \\ x(u_0) & x(u_1) & x(v_0) & x(v_1) \\ y(u_0) & y(u_1) & y(v_0) & y(v_1) \\ z(u_0) & z(u_1) & z(v_0) & z(v_1) \end{vmatrix} = 0$$

The substitution of the y- and z-coordinates of each vertex in the equation above gives a condition on the x-coordinates of the vertices that must be satisfied in order to have a crossings between e_0 and e_1. Thus it is sufficient to prove that the equation has no solution in order to prove that e_0 and e_1 do not cross each other. We call this equation *co-planarity equation* of e_0 and e_1.

A *queue layout* [14, 15] of a graph G consists of a linear ordering λ of the vertices of G, and a partition of the edges of G into queues, such that no two edges in the same queue are *nested* with respect to λ. In other words there are no edges $e_0 = (u_0,v_0)$ and $e_1 = (u_1,v_1)$ such that e_0 and e_1 are assigned to the same queue and $u_0 < u_1 < v_1 < v_0$ in λ. A queue layout with q queues is also called a *q-queue layout*. The *queue number* of a graph G, denoted by $qn(G)$, is the minimum q such that G has a q-queue layout.

2.2 Previous Results

In this section we recall some previous results about track number and the relations between the track number and the queue number of a graph.

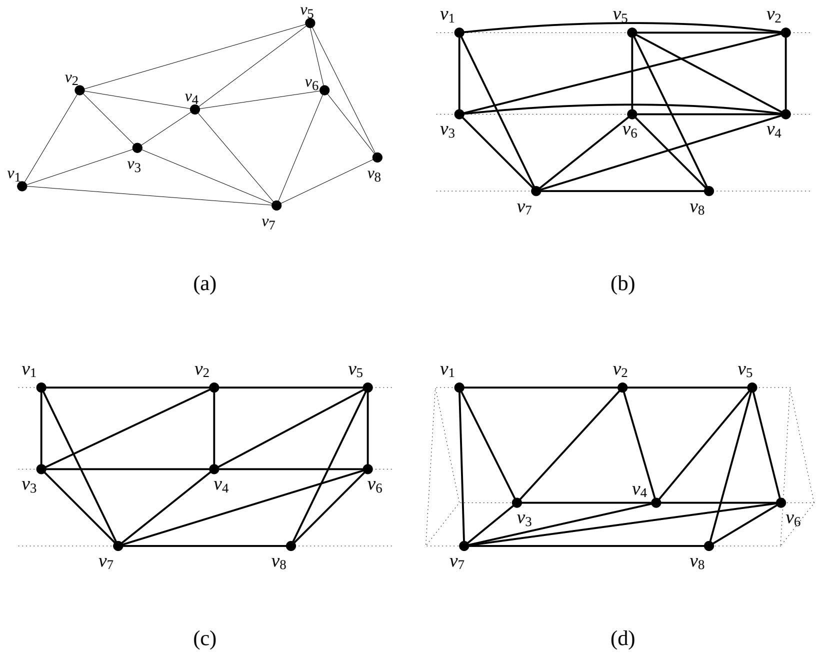

Figure 1: (a) A graph G. (b) A track assignment of G. (c) A track layout of G. (d) A track drawing of G.

In [4] the graphs having track number 2 are characterized, and it is proved that they are a subclass of the outerplanar graphs. The following lemma is an immediate corollary of the result in [4].

Lemma 1 *[4] The class of graphs that admit a 2-track layout is a subclass of the outerplanar graphs.*

In [10] it is proved that every outerplanar graph admits a track drawing on 3 tracks, and hence has track number 3. Also, it is proved that there exist trees (and hence outerplanar graphs) that do not admit a track layout on 2 tracks. The following hold.

Theorem 2 *[10] Every outerplanar graph has track number at most 3 and admits a track drawing with volume at most $2 \times 2 \times n$. Also, there exists an outerplanar graph G such that $tn(G) \geq 3$.*

In [7] the relation between a track layout and a track drawing is studied. The following theorem holds.

Theorem 3 *[7] Let G be a graph with n vertices such that $tn(G) = t$. Then:*

- *G admits a t-track drawing whose volume is $t \times p_t \times p_t \cdot n'$;*
- *G admits a $2t$-track drawing whose volume is $2t \times p_{2t} \times p_{2t} \cdot \lceil \frac{n}{t} \rceil$;*

where p_t is the smallest prime number greater than t, p_{2t} is the smallest prime number greater than $2t$ and n' is the maximum number of vertices on a single track.

Also, in [18] the track number of a graph is related to the queue number of a graph. In particular the following holds.

Theorem 4 *[18] Let G be a graph with star chromatic number $\chi_{st}(G) \leq c$, and queue number $qn(G) \leq q$. Then G has a t-track layout, where $t \leq c(2(c-1)q+1)^{c-1}$.*

3 Arched Leveled Planar Graphs

In this section we will study the track number of graphs with queue number equal to 1. These graphs are characterized in [15], where it is shown that they are planar graphs and admit a leveled planar embedding. For this reason they are called *arched leveled planar graphs*. We first give a lower bound on the track number and then we present an upper bound that improves the result in [18]. We show that the track number of an arched leveled planar graph is at most 5. Also, we prove that there exists an arched leveled planar graph whose track number is at least 4. We start with a basic lemma that is used to prove the lower bound.

Lemma 5 *Let G_0 be the graph in Figure 2(a). Then in any* 3*-track layout, vertices u and v are on different tracks, and at least one of the vertices w_i ($i = 0,\ldots,3$) is on the third track.*

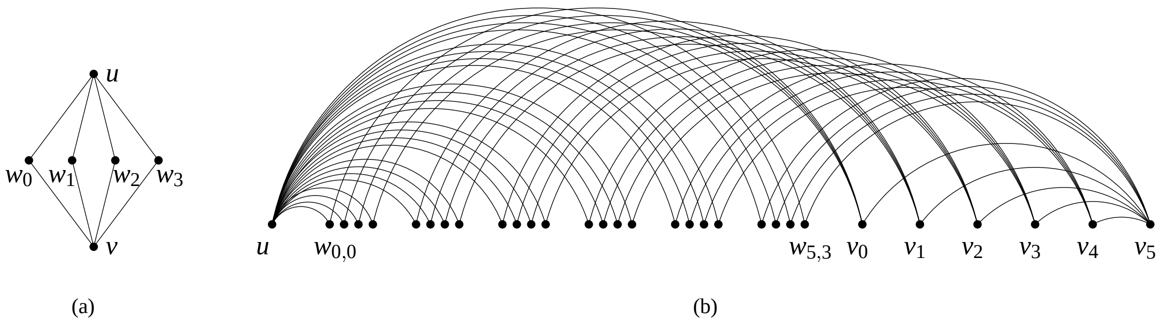

Figure 2: (a) Graph G_0 of Lemma 5. (b) Graph G with $qn(G) = 1$ and $tn(G) = 4$.

Lemma 6 *Let G be the graph in Figure 2(b). Then $qn(G) = 1$ and $tn(G) \geq 4$.*

Sketch of Proof. The proof that G has queue number 1 follows from Figure 2(b) where a 1-queue layout of G is given. We prove that it cannot be laid out on 3 tracks. Suppose for a contradiction that there exists a 3-track layout of G with tracks t_0, t_1 and t_2, and let t_0 be the track containing u. By Lemma 5 vertices v_i ($0 \leq i \leq 5$) must be on a track different from t_0. Let t_1 be the track containing v_5. Two cases are possible:

1. There exist three vertices v_i, v_j and v_k ($0 \leq i,j,k \leq 4$) that are in t_1. In this case two of the three edges (v_5,v_i), (v_5,v_j), and (v_5,v_k) form an overlap.
2. There exist three vertices v_i, v_j and v_k, that are in t_2. By Lemma 5, there exist three vertices $w_{i,a}$, $w_{j,b}$ and $w_{k,c}$ ($0 \leq a,b,c \leq 3$) in track t_1 adjacent to u and to v_i, v_j and v_k respectively. Assume without loss of generality that $v_i < v_j < v_k$ in t_2. It follows that $w_{i,a} < w_{j,b} < w_{k,c}$ in t_1, because otherwise there would be an X-crossing between edges $(v_i,w_{i,a})$, $(v_j,w_{j,b})$ and $(v_k,w_{k,c})$. If $v_5 < w_{i,a}$ then edges (v_5,v_j) and $(v_i,w_{i,a})$ form an X-crossing. If $w_{i,a} < v_5 < w_{j,b}$ then edges (v_5,v_k) and $(v_j,w_{j,b})$ form an X-crossing. If $w_{j,b} < v_5 < w_{k,c}$ then edges (v_5,v_i) and $(v_j,w_{j,b})$ form an X-crossing. Finally, if $v_5 > w_{k,c}$ then edges (v_5,v_j) and $(v_k,w_{k,c})$ form an X-crossing.

It follows that $tn(G) \geq 4$. ■

In order to prove the upper bound on the track number of arched leveled planar graphs we first describe a block decomposition of a connected graph G with $qn(G) = 1$. Then we will describe how to assign the vertices to 5 tracks. Suppose we have a 1-queue layout of G where the linear ordering of the vertices is $\lambda = v_0,v_1,\ldots,v_{n-1}$. We say that $v_i < v_j$ if $i < j$. A vertex v is called a *special cut vertex* if there are no edges (u,w) with $u < v < w$. A block is a subset of consecutive vertices in λ. The first vertex of a block B_i is called *source vertex* and is denoted as s_i. The last vertex of a block B_i is called *sink vertex* and is denoted as t_i. All the other vertices are called *internal vertices* of B_i. The *block decomposition* is defined as follows

- Block B_0 consists of the vertices $v_0,v_1,\ldots,v_j$, where v_j is such that there exists the edge (v_0,v_j) and there is no edge (v_0,v_h) with $v_j < v_h$. In other words j is the largest index such that there is an edge (v_0,v_j).
- If the sink vertex t_{i-1} of block B_{i-1} is not a special cut vertex then block B_i consists of the vertices $v_k,v_{k+1},\ldots,v_j$ with the following properties: there is an edge (v_k,v_j); $v_k < t_{i-1}$; there is no edge (v_g,v_h) with $v_g < t_{i-1}$ and $v_j < v_h$; there is no edge (v_h,v_j) with $v_h < v_k$. In other words j is the largest index such that there is an edge from a vertex smaller than t_{i-1} to v_j and k is the smallest index such that there is an edge (v_k,v_j).
- If the sink vertex t_{i-1} of block B_{i-1} is a special cut vertex then block B_i consists of the vertices $v_k,v_{k+1},\ldots,v_j$, where $v_k = t_{i-1}$, and where v_j is such that there exists an edge (t_{i-1},v_j) and there is no edge (t_{i-1},v_h) with $h > j$. In other words j is the largest index such that there is an edge from t_{i-1} to v_j.

An example of a block decomposition is illustrated in Figure 3.

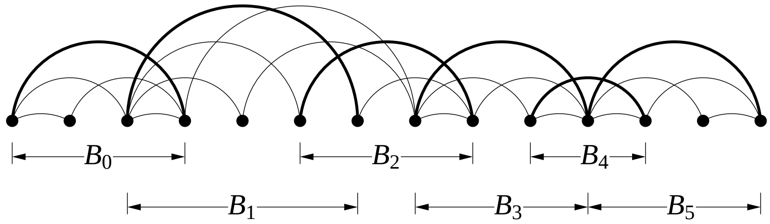

Figure 3: A block decomposition of an arched leveled planar graph.

Algorithm ALPTRACKLAYOUT() computes a 5-track layout of an arched leveled planar graph G.

ALPTRACKLAYOUT(G)
  Input: 1-queue layout of connected graph G with $\lambda = v_0, v_1, \ldots, v_{n-1}$
  Output: A 5-track layout of G.
Let 0,1,2,3 and 4 be five tracks;
Let $B_0, \ldots, B_{k-1}$ be a block decomposition of G;
$track(s_0) \leftarrow 0$;
for $i = 0$ **to** $k-1$
    $t \leftarrow track(s_i)$;
    foreach internal vertex v of B_i not assigned to any track
        $track(v) \leftarrow (t+1) \mod 5$;
    endfor
    $track(t_i) \leftarrow (t+2) \mod 5$;
endfor
for $i = 0$ **to** $n-1$
    $order(v_i) = i$;
endfor

Algorithm 1: Algorithm ALPTRACKLAYOUT()

Lemma 7 *Let G be a connected graph with $qn(G) = 1$. Let $\lambda = v_0, v_1, \ldots, v_{n-1}$ be the linear ordering of a 1-queue layout of G. The track assignment computed by Algorithm* ALPTRACKLAYOUT*() is such that for all edges (v_g, v_h) with $g < h$*

$$track(v_h) = (track(v_g) + 1) \mod 5 \qquad or$$
$$track(v_h) = (track(v_g) + 2) \mod 5.$$

Lemma 8 *Let G be a graph with $qn(G) = 1$, then $tn(G) \leq 5$.*

Sketch of Proof. We first assume that G is connected. Suppose we have a 1-queue layout of G with $\lambda = v_0, v_1, \ldots, v_{n-1}$. We prove that Algorithm ALPTRACKLAYOUT() computes a track layout of G. We have no overlaps since there are no edges (v_i, v_j) with $track(v_i) = track(v_j)$. Also we have no X-crossing. Consider two edges (v_g, v_h) and (v_i, v_j) such that v_g and v_i are in track a and v_h and v_j are in another track b. By Lemma 7, there are two cases: (i) if $b = (a+1) \mod 5$ or $b = (a+2) \mod 5$, then $v_g < v_h$ and $v_i < v_j$; (ii) if $b = (a-1) \mod 5$ or $b = (a-2) \mod 5$, then $v_g > v_h$ and $v_i > v_j$. Consider case (i), and assume without loss of generality that $v_g < v_i$. If $v_j < v_h$ then $v_g < v_i < v_j < v_h$ in λ, but this is not possible, because there would be two nested edges in the 1-queue layout. Hence $v_h < v_j$ and therefore there is no X-crossing. Consider case (ii), and assume again that $v_g < v_i$. If $v_j < v_h$ then it would be $v_j < v_h < v_g < v_i$ in λ, but this is not possible, because there would be two nested edges in the 1-queue layout. Hence $v_h < v_j$ and therefore there is no X-crossing. If G is disconnected, we can apply Algorithm ALPTRACKLAYOUT() to each component of G. ■

Lemma 8 proves that every arched leveled planar graph G has track number at most 5. By Theorem 3 G admits a track drawing whose volume is $10 \times 11 \times 11 \times \lceil \frac{n}{5} \rceil$. It is possible to reduce this volume by an ad-hoc drawing technique. The following theorem summarize the results above and gives a reduced bound on the volume of the track drawing.

Theorem 9 *Every arched leveled planar graph has track number at most 5 and admits a track drawing with volume at most* $3 \times 3 \times n$ *that can be computed in* $O(n)$ *time. Also, there exists an arched leveled planar graph* G *such that* $tn(G) \geq 4$.

Sketch of Proof. The bounds on the track number follow from Lemma 6 and 8. We describe now how to compute a track drawing with volume $3 \times 3 \times n$. Consider a restricted integer grid ϕ consisting of the five track lines $(x,2,1)$, $(x,0,1)$, $(x,0,0)$, $(x,2,0)$ and $(x,1,2)$ and number these track lines 0,1,2,3, and 4, respectively. Compute a track layout with five tracks using Algorithm ALPTRACKLAYOUT(). Let n_0, n_1, n_2, n_3 and n_4 be the number of vertices in track 0, 1, 2, 3 and 4, respectively. Draw the vertices assigned to track i on track line i according to the total order defined in the track, so that they occupy x-coordinates from $\sum_{j=0}^{i-1} n_j$ to $\sum_{j=0}^{i} n_j$ $(i = 0,1,2,3,4)$. We prove that the drawing has no crossing. Overlaps and X-crossings are not possible, because there is no overlap and no X-crossing in the track layout.

A crossing is possible only between edges that are on two different strips crossing each other along a straight-line which is not one of the five track line 0,1,2,3, and 4. There are five pairs of such strips: (1) σ_{02} and σ_{13}; (2) σ_{01} and σ_{24};(3) σ_{01} and σ_{34}; (4) σ_{02} and σ_{34};(5) σ_{13} and σ_{24}. Let σ_{ij} and σ_{hk} be the two crossing strips of one of the five cases above and let e_1 and e_2 be two edges on σ_{ij} and σ_{hk}, respectively. Denote as x_i, x_j, x_h and x_k the x-coordinates of the vertices on track lines i, j, h and k. The co-planarity equations for each of the five cases are:

$$\begin{array}{ll}
(1) & x_0 + x_2 = x_1 + x_3 \\
(2) & x_0 + 3x_1 = 2x_2 + 2x_4 \\
(3) & 3x_0 + x_1 = 2x_3 + 2x_4 \\
(4) & 4x_0 + x_2 = 3x_3 + 2x_4 \\
(5) & 4x_1 + x_3 = 3x_2 + 2x_4
\end{array}$$

Since $x_0 < x_1 < x_2 < x_3 < x_4$, none of the above equations has a solution, and therefore no crossing is possible.

The drawing algorithm consider a vertex per time and executes a constant number of operations for each vertex. The time complexity is then $O(n)$. ■

Theorem 9 shows that every arched leveled planar graph admits a track layout on 5 tracks. However, there are specific classes of graphs that have queue number 1 and that admit a track layout on less than 5 tracks. Trees and unicyclic graphs are known to have queue number 1 [15]. A *unicyclic graph* is a graph such that each connected component contains at most one cycle. The family of unicyclic graphs includes trees. Since both trees and unicyclic graphs are outerplanar, by [10] they admit a track layout on 3 tracks with volume $2 \times 2 \times n$. In [10], it is also shown that there exist trees (and hence unicyclic graphs) that cannot be laid out on two tracks.

Another class of graphs having queue number 1 are the square meshes [15]. An $a \times b$ *square mesh* is a graph with vertex set $V = \{v_{ij} \mid 0 \leq i < a, 0 \leq j < b\}$ and edge set $E = \{(v_{ij}, v_{i,j+1}) \mid 0 \leq j < b-1\} \cup \{(v_{ij}, v_{i+1,j}) \mid 0 \leq i < a-1\}$.

A lower bound on the track number of the square meshes is 3. Since there exist square meshes that are not outerplanar and by Lemma 1 they cannot be laid out on two tracks.

On the other side square meshes can be easily laid out on three tracks as follows. Let G be an $a \times b$ square mesh $(a \geq b)$. Set $track(v_{ij}) = i \mod 3$ and $order(v_{ij}) = (ia + j)$ $(0 \leq i < a, 0 \leq j < b)$. The track assignment is clearly without X-crossings and overlaps and hence it is a 3-track layout. By Theorem 3, a square meshe admits a track drawing with volume $3 \times 5 \times 5n'$, where n' is the maximum number of vertices on a single track. We will show in the proof of next Theorem that $n' = \frac{n+2\sqrt{n}}{3}$ and therefore the volume is $3 \times 5 \times 5 \times \frac{n+2\sqrt{n}}{3}$. This volume can be reduced by an ad-hoc drawing technique. The following Theorem holds.

Theorem 10 *Every square mesh has track number at most 3 and admits a track drawing with volume at most* $2 \times 2 \times \frac{n+2\sqrt{n}}{3}$ *that can be computed in* $O(n)$ *time. Also, there exists a square mesh* G *such that* $tn(G) \geq 2$.

Sketch of Proof. A track drawing with volume less than the one given by Theorem 3, can be computed as follows. Consider a restricted integer grid consisting of the three track lines $(x,0,0)$, $(x,1,0)$ and $(x,0,1)$ and compute a track layout with three tracks as described above. Let n_0, n_1 and n_2 be the number of vertices on track 0, 1 and 2, respectively. Draw the vertices assigned to track i on track line i according to the total order defined on the track, so that they occupy x-coordinates from 0 to $n_i - 1$ $(i = 0,1,2)$. Overlaps and an X-crossings are not possible because there is no overlap and no X-crossing in the track layout. No other crossings is possible because there are no two strips whose intersection is different from one of the three track lines. The volume of the obtained drawing is $2 \times 2 \times \max\{n_0, n_1, n_2\}$. The maximum among n_0, n_1 and n_2 is given by $\lceil \frac{a}{3} \rceil \cdot b$. We have

$$\left\lceil \frac{a}{3} \right\rceil b \leq \frac{a+2}{3} b = \frac{ab+2b}{3} \leq \frac{n+2\sqrt{n}}{3}.$$

Therefore the volume is bounded by $2 \times 2 \times \frac{n+2\sqrt{n}}{3}$. The drawing algorithm described above consider a vertex per time and for each vertex v computes the two values of $track(v)$ and $order(v)$. The time complexity is clearly $O(n)$. ■

4 X-trees

An *X-tree* is a complete ordered binary tree with some extra edges connecting vertices at the same level. More precisely, for each level of the tree, if $v_0, v_1, \ldots, v_{k-1}$ are the vertices of that level in the left-to-right order, the extra edges are (v_i, v_{i+1}) $(0 \leq i \leq k-2)$. The X-trees have queue number 2 [15]. A lower bound on the track number of X-trees is trivially 3, since there exist X-trees that are not outerplanar and therefore, by Lemma 1 cannot be laid out on 2 tracks. Three is, in fact, also an upper bound for the track number of X-trees. A 3-track layout of an X-tree G can be easily computed by laying out the complete binary tree underlying G according to the algorithm in [10]. The algorithm consists of a BFS visit of the tree. For each visited vertex v we set $track(v) = d \mod 3$ and $order(v) = n_v + 1$, where d is the distance of v from the root of the tree and n_v is the number of vertices visited before v. The track assignment obtained for the underlying tree is without overlaps and X-crossings as proved in [10]. If the BFS visit is performed so that the children of each vertex are visited according to their left-to-right order then the vertices of each extra edge are consecutive in the same track. It follows that the extra edges do not introduce any overlap or X-crossing. By Theorem 3, an X-tree admits a track drawing with volume $3 \times 5 \times 5n'$, where n' is the maximum number of vertices on a single track. We will show in the proof of next Theorem that $n' = \frac{4}{7}(n+1)$ and therefore the volume is $3 \times 5 \times 5 \times \frac{4}{7}(n+1)$. This volume can be reduced by an ad-hoc drawing technique. The following Theorem holds.

Theorem 11 *Every X-tree has track number at most 3 and admits a track drawing with volume at most* $2 \times 2 \times \frac{4}{7}(n+1)$ *that can be computed in $O(n)$ time. Also, there exists an X-tree G such that $tn(G) \geq 2$.*

Sketch of Proof. We describe how to compute a track drawing with volume $2 \times 2 \times \frac{4}{7}(n+1)$. Consider a restricted integer grid ϕ consisting of the three track lines $(x,0,0)$, $(x,1,0)$ and $(x,0,1)$ and number these track lines 0,1, and 2, respectively. Compute a track layout on three tracks as described above. Let n_0, n_1 and n_2 be the number of vertices on track 0, 1 and 2, respectively. Draw the vertices assigned to track i on track line i according to the total order defined on the track, so that they occupy x-coordinates from 0 to $n_i - 1$ $(i = 0,1,2)$. Overlaps and X-crossings are not possible because there is no overlap and no X-crossing in the track layout. No other crossings is possible because there are no two strips whose intersection is different from one of the three track lines. The volume of the obtained drawing of the X-tree G is $2 \times 2 \times \max\{n_0, n_1, n_2\}$. We have:

$$\begin{aligned} n_0 &= 2^0 + 2^3 + 2^6 + \ldots \\ n_1 &= 2^1 + 2^4 + 2^7 + \ldots \\ n_2 &= 2^2 + 2^5 + 2^8 + \ldots \end{aligned}$$

Let d be the maximum depth of G. The number of leaves of the complete binary tree underlying G is 2^d. We have $d = \log_2(n+1) - 1$ and hence $2^d = (n+1)/2$. It follows that the track line with the maximum number of vertices is the one where the leafs lie. Therefore the maximum among n_0, n_1 and n_2 is:

$$n_{max} = 2^d + 2^{d-3} + 2^{d-6} + \cdots + 2^b \text{ with } b = 0 \text{ or } 1 \text{ or } 2$$

We have:

$$n_{max} \leq 2^d(1 + 2^{-3} + 2^{-6} + 2^{-9} + \ldots) = 2^d \sum_{i=0}^{\infty} 2^{-3i} = 2^d \frac{1}{1-2^{-3}} = 2^d \frac{8}{7} = (n+1)\frac{4}{7}.$$

Therefore the volume is bounded by $2 \times 2 \times \frac{4}{7}(n+1)$. The drawing algorithm consists of a BFS visit and therefore the time complexity is $O(n)$. ■

5 Halin Graphs

In this section we study the track number of a well-investigated family of graphs called Halin Graphs [13]. A *Halin graph* is a graph such that:

- every vertex of G has degree greater or equal to 3;
- G can be decomposed into a spanning tree T of G and a cycle C through the leaves of T;
- G has a planar embedding in which C is the boundary of the external face.

T is called the *characteristic tree* of G and C is called the *adjoint cycle* of G. Figure 4 shows a Halin graph.

It is known from the existing literature [11] that 3 queues are always sufficient for a queue layout of a Halin graph and that 2 queues are sometimes necessary. A lower bound on the track number of Halin graphs is 3, since Halin graphs are not outerplanar (every outerplanar graph has at least on vertex of degree two) and therefore, by Lemma 1 cannot be laid out on 2 tracks.

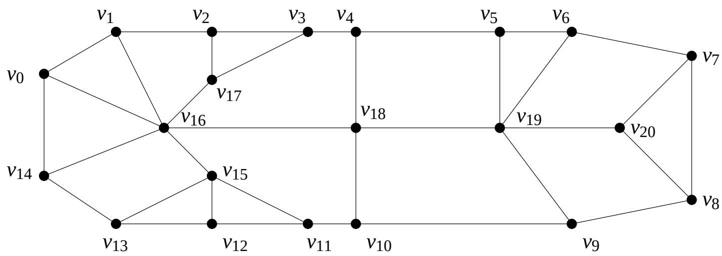

Figure 4: A Halin graph.

We now describe an algorithm to compute a 4-track layout of a Halin graph. An *external path* of an embedded rooted ordered tree T is the path $\pi = \pi_l \cup \pi_r$, where π_l is the path from the leftmost leaf of T to the root of T and π_r is the path from the rightmost leaf of T to the root of T. Let v be a vertex in an external path π. If v has children that are not in π then every subtree rooted at a child of v not in π is called a *dangling subtree* of π.

Let G be a Halin graph. Assume that G is embedded in the plane such that it is planar and its adjoint cycle C is the exterior face. Let T be the characteristic tree of G. T inherits its embedding from G. Arbitrarily choose one of the non-leaf vertices of T as the root. A *level decomposition* of T is an assignment of a level to each vertex v of T that is defined as follows (see Figure 5): (i) all the vertices on the external path of T are given level 0; (ii) Let π be an external path of level i. For any dangling subtree T' of π, the vertices on the external path of T' are given level $i+1$. Let π be an external path of any level. Let $T_0, T_1, \ldots, T_{h-1}$ be the dangling subtrees of π. We define a *natural ordering* of the dangling subtrees as follows: (i) $T_0, T_1, \ldots, T_{h-1}$ are ordered from left to right according to their parents order in π; (ii) dangling subtrees that have the same parent are ordered from left to right.

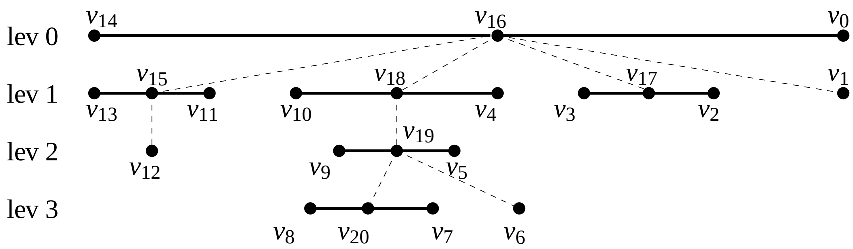

Figure 5: A level decomposition of the characteristic tree of the Halin graph in Figure 4.

A consequence of the level decomposition is the following.

Property 12 *Let G be a Halin graph and let π be an external path of any level with at least one dangling subtree. Let $T_0, T_1, \ldots, T_{h-1}$ be the natural order of the dangling subtrees of π. Then in the adjoint cycle C of G:*

1. *the leftmost leaf of π is adjacent to the leftmost leaf of T_0;*
2. *the rightmost leaf of π is adjacent to the rightmost leaf of T_{h-1};*

3. the rightmost leaf of T_j is adjacent to the leftmost leaf of T_{j+1} ($j = 0, \ldots, h-2$).

Since the embedding of G is not changed when T is rooted, and since the boundary of the external face of G is C, then there exist an edge of C that connects the leftmost leaf of T to the rightmost leaf of T. The edge connecting the leftmost leaf of T and the rightmost leaf of T is called the *long edge*. Let π be an external path without dangling subtrees; if π consists of three vertices u, v and w in this order and u and w are leafs of T, then u and w are connected by an edge of C and this edge is called an *overlapping edge*.

We now describe an algorithm to compute a 3-track layout of a Halin graph without its long edge and overlapping edges. Later we will describe how the long edge and the overlapping edges can be added back to the track layout using a fourth track. A Halin graph after the deletion of the long edge and the overlapping edges is called a *reduced Halin graph*.

```
RHTrackLayout(G)
  Input: An embedded reduced Halin graph G
  Output: A track layout of G on the 3 tracks 0,1,2
Let T be the characteristic tree of G;
Root T at any vertex r;
Q ← new queue();
Q.enqueue(r);
ord ← 0;
while Q is not empty
    v ← Q.dequeue();
    if v = r
        t ← 0;
    else
        t ← (track(parent(v)) + 1)  mod 3;
    endif
    Let T' be the subtree rooted at v;
    Let π = v_0, v_1, ..., v_{h-1} be the external path of T';
    for i = 0 to h - 1
        track(v_i) ← t;
        order(v_i) ← ord;
        ord ← ord + 1;
        Let w_0, w_1, ..., w_{k-1} be the children of v_i not in π ordered from left to right;
        for j = 0 to k - 1
            Q.enqueue(w_j);
        endfor
    endfor
endwhile
```

Algorithm 2: Algorithm RHTrackLayout()

Lemma 13 *Let G be a reduced Halin graph. Algorithm* RHTrackLayout() *computes a 3-track layout of G.*

Sketch of Proof. We prove that the track assignment computed by Algorithm RHTrackLayout() has no overlaps nor X-crossing. The edges having both vertices in a track are either edges of an external path or edges of the adjoint cycle connecting leafs of the same level. The edges of the external paths do not overlap since the two vertices of each edge are consecutive in a track. Let π be an external path of level i and let $T_0, T_1, \ldots, T_{h-1}$ be the natural ordering of the dangling subtrees of π. By Property 12 the edges of the adjoint cycle connecting two leafs of the same level $i+1$ connect the rightmost leaf of T_j to the leftmost leaf of T_{j+1}. Since Algorithm RHTrackLayout() lays out $T_0, T_1, \ldots, T_{h-1}$ according to their natural order, then the rightmost leaf of T_j and the leftmost leaf of T_{j+1} are consecutive in a track and therefore they do not overlap.

Let $e_0 = (u_0, v_0)$ and $e_1 = (u_1, v_1)$ be two edges having the two vertices in two different tracks and assume that u_0 and u_1 are in the same track with $u_0 < u_1$. Edges e_0 and e_1 are either edges connecting the roots of dangling trees to their parents or edges of the adjoint cycle connecting leafs at consecutive levels. If u_0 and u_1 are in different external path π_0 and π_1, then u_0 is adjacent to a vertex (a leaf or the root) of a dangling subtree of π_0 and u_1 is adjacent to a vertex (a leaf or the root) of a dangling subtree of π_1. The vertices of the dangling subtrees of π_0 precede the vertices of the dangling subtree of π_1 in each track. Therefore $v_0 < v_1$ and an X-crossing is not possible. Assume u_0 and u_1 be in the same external path π. If u_0 is the leftmost vertex of π then v_0 is the leftmost leaf of the first dangling subtree T_0 of π (Property 12). It follows that $v_0 < v_1$ and an X-crossing is not possible also in this case. If u_1 is the rightmost vertex of π then v_1 is the is the rightmost leaf of the last dangling subtree T_{h-1} of π (Property 12). It follows that $v_0 < v_1$ and an X-crossing is not possible. If u_0 is not the leftmost vertex of π and u_1 is not the rightmost vertex of π, then they are adjacent to the roots of two dangling subtrees of π. Since $u_0 < u_1$ and since the dangling subtree are laid out according to their natural order, then $v_0 < v_1$ and an X-crossing is not possible. ■

Notice that the track layout of the Reduced Halin Graph is such that the vertices of each removed edge (long edge or overlapping edge) are both in the same track. The track layout of the reduced Halin graph of the graph in Figure 4 is shown in Figure 6. The long edge (v_0, v_{14}) and the overlapping edges (v_2, v_3) and (v_7, v_8) would create an overlap if considered in the track layout.

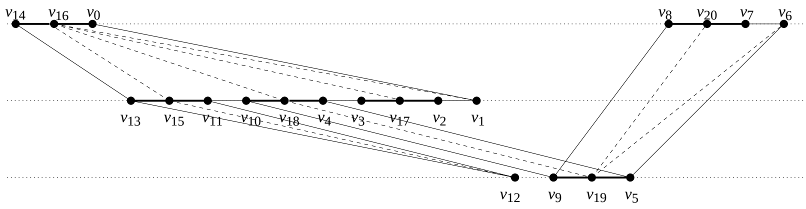

Figure 6: A track layout of the reduced Halin graph of the graph in Figure 4.

Lemma 14 *Let G be a Halin graph and let G' the corresponding reduced Halin graph. Let $\Gamma(G')$ be a 3-track layout of G' computed by Algorithm* RHTRACKLAYOUT*() and let $e_0 = (u_0, v_0)$ and $e_1 = (u_1, v_1)$ be any pair of edges in $G - G'$. The two edges e_0 and e_1 do not have any vertex in common, and if their vertices are in the same track then either $u_0 < u_1$ and $v_0 < v_1$ or $u_1 < u_0$ and $v_1 < v_0$.*

By Lemma 14 it is easy to see that a track layout of a Halin graph can be computed from a 3-track layout of a reduced Halin graph by adding a new track and moving the left vertex of each overlapping edge and the left vertex of the long edge to the new track.

Lemma 15 *Let G be a Halin graph, then $tn(G) \leq 4$.*

Sketch of Proof. Let $\Gamma(G)$ the 3-track layout of the reduced Halin graph G' of G computed by Algorithm RHTRACKLAYOUT(). Denote as 0,1, and 2 the three track used and consider a new track denoted as 4. For each edge $e = (u, v)$ in $G - G'$ assume that $u < v$ and set $track(v) = 4$, i.e. for each edge in $G - G'$ one of the vertices is assigned to the new track. This change of track do not introduce an overlap since no edge has both vertices on track 4. Also, no X-crossing is introduced. Namely, consider the the pair of tracks consisting of track 4 and of any of the other three tracks (denote this track as i). The edges whose vertices are in this pair are the overlapping edge (and/or the long edge) whose vertices were originally in i. Let $e_0 = (u_0, v_0)$ and $e_1 = (u_1, v_1)$ be two of these edges and assume that v_0 and v_1 are in track 4 with $u_0 < u_1$. By Lemma 14 we have $u_0 < v_0 < u_1 < v_1$ on track i in $\Gamma(G')$. Since $order(u_0)$, $order(v_0)$, $order(u_1)$, and $order(v_1)$ are not changed, we have $u_0 < u_1$ in track 4 and $v_0 < v_1$ in track i. So there is no X-crossing. ■

By Lemma 15 the track number of a Halin graph is at most 4. By the results in [7] this is sufficient to say that G admits a track drawing with volume $O(n)$. In particular, by Theorem 3 an upper bound on the volume is $4 \times 5 \times 5n'$. We describe now how this volume can be reduced to $2 \times 2 \times n$.

Theorem 16 *Every Halin graph has track number at most 4 and admits a track drawing with volume at most $2 \times 2 \times n$ that can be computed in $O(n)$ time. Also, for every Halin graph $tn(G) \geq 3$.*

Sketch of Proof. The results about the bounds on track number follows from the fact that Halin graphs are not outerplanar and by Lemma 15. We prove the result about volume. Consider a restricted integer grid ϕ consisting of the four track lines $(x, 0, 0)$, $(x, 1, 0)$, $(x, 0, 1)$ and $(x, 1, 1)$, and number this track lines 0,1,2, and 3, respectively. Compute a track layout on four tracks as described above . Let n_0, n_1, n_2 and n_3 be the number of vertices on track 0, 1, 2 and 3, respectively. Draw the vertices assigned to track i on track line i according to the total order defined on the track, so that they occupy x-coordinates from $\sum_{j=0}^{i-1} n_j$ to $\sum_{j=0}^{i} n_j$ $(i = 0, 1, 2, 3, 4)$. We prove that the drawing has no crossing. Overlap and X-crossings are not possible, because there is no overlap and no X-crossing in the track layout.

A crossing is then possible only between edges on two different strips that cross each other along a straight line that is not one of the four track lines 0,1,2 and 3. There are only two such strips σ_{02} and σ_{13}. The co-planarity equation of an edge on σ_{02} and an edge on σ_{13} is:

$$x_1 + x_3 = x_0 + x_2$$

Since $x_0 < x_1 < x_2 < x_3$ then the equation has no solution, i.e. a crossing is not possible. It follows that the drawing is a track drawing with volume $2 \times 2 \times n$. The drawing algorithm consider a vertex per time and executes a constant number of operations for each vertex. The time complexity is therefore $O(n)$. ■

6 Open Problems

The general question of computing 3D straight line grid drawing of minimum volume still remains largely unexplored. Some questions that might help to better understand this problem naturally raise from the work in this paper.

- To reduce the gap between the lower bound of four and the upper bound of five for the track number of Arched Leveled Planar graphs and of Halin Graphs.
- To find upper and lower bounds on the track number of other families of graphs.
- To find new algorithms that compute drawings with linear volume and better aspect ratio. Namely, the volume of the drawings computed by the algorithms presented in this paper is $O(n)$. This is obtained at expense of an aspect ratio that is also $O(n)$. It would be interesting to find new drawing technique that could obtain a linear volume and a good aspect ratio at the same time.

Acknowledgments

We thank Giuseppe Liotta for useful discussions on the subject of this paper.

References

[1] T. Calamoneri and A. Sterbini. Drawing 2-, 3- and 4-colorable graphs in $o(n^2)$ volume. In S. North, editor, *Graph Drawing (Proc. GD '96)*, volume 1190 of *Lecture Notes Comput. Sci.*, pages 53–62. Springer-Verla, 1997.

[2] M. Chrobak, M. T. Goodrich, and R. Tamassia. Convex drawings of graphs in two and three dimensions (preliminary version). In *Proceedings of the twelfth annual symposium on Computational geometry*, pages 319–328. ACM Press, 1996.

[3] R. F. Cohen, P. Eades, T. Lin, and F. Ruskey. Three-dimensional graph drawing. *Algorithmica*, 17(2):199–208, 1997.

[4] S. Cornelsen, T. Schank, and D. Wagner. Drawing graphs on two and three lines. In M. Goodrich and S. Kobourov, editors, *Graph Drawing (Proc. GD '02)*, volume 2528 of *Lecture Notes Comput. Sci.*, pages 31–41. Springer-Verlag, 2002.

[5] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing.* Prentice Hall, Upper Saddle River, NJ, 1999.

[6] E. Di Giacomo, G. Liotta, and S. K. Wismath. Drawing series-parallel graphs on a box. In *Proc. of the 14th Canadian Conference on Computational Geometry (CCCG), University of Lethbridge, Alberta, Canada*, 2002.

[7] V. Dujmovi´c, P. Morin, and D. Wood. Pathwidth and three-dimensional straight line grid drawings of graphs. In M. T. Goodrich and S. Koburov, editors, *Graph Drawing (Proc. GD '02)*, volume 2528 of *Lecture Notes Comput. Sci.*, pages 42–53. Springer-Verlag, 2002.

[8] V. Dujmovi´c and D. Wood. Tree-partitions of k-trees with application in graph layout. Technical Report TR-02-03, School of Computer Science, Carleton University, 2002.

[9] V. Dujmovi´c and D. Wood. Tree-partitions of k-trees with application in graph layout. In *Workshop on Graph Theoretic Concepts in Computer Science*, Lecture Notes Comput. Sci. Springer-Verlag, to appear.

[10] S. Felsner, G.Liotta, and S. Wismath. Straight line drawings on restricted integer grids in two and three dimensions. In P. Mutzel, M. Junger, and S. Leipert, editors, *Graph Drawing (Proc. GD '01)*, volume 2265 of *Lecture Notes Comput. Sci.* Springer-Verlag, 2001.

[11] J. L. Ganley. Stack and queue layouts of halin graphs. 1995, manuscript.

[12] A. Garg, R. Tamassia, and P. Vocca. Drawing with colors. In *Proc. 4th Annu. European Sympos. Algorithms*, volume 1136 of *Lecture Notes Comput. Sci.*, pages 12–26. Springer-Verlag, 1996.

[13] R. Halin. Studies in minimally connected graphs. In D. J. A. Welsh, editor, *Combinatorial Mathematics and its Applications*, pages 129–136. New York, Academic Press, 1971.

[14] L. S. Heath, F. T. Leighton, and A. L. Rosenberg. Comparing queues and stacks as mechanisms for laying out graphs. *SIAM J. Disc. Math.*, 5(3):398–412, 1992.

[15] L. S. Heath and A. L. Rosenberg. Laying out graphs using queues. *SIAM J. Computing*, 21:927–958, 1992.

[16] M. Kaufmann and D. Wagner, editors. *Drawing Graphs*, volume 2025 of *Lecture Notes in Computer Science*. Springer-Verlag, 2001.

[17] J. Pach, T. Thiele, and G. Tóth. Three-dimensional grid drawings of graphs. In G. Di Battista, editor, *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes Comput. Sci.*, pages 47–51. Springer-Verlag, 1997.

[18] D. Wood. Queue layouts, tree-width, and three-dimensional graph drawing. In M. Agrawal and A. Seth, editors, *Proc. 22nd Foundations of Software Technology and Theoretical Computer Science (FSTTCS '02)*, volume 2556 of *Lecture Notes Comput. Sci.*, pages 348–359. Springer-Verlag, 2002.

Appendix

Proof of Lemma 5

Lemma 5 *Let G_0 be the graph in Figure 2(a). Then in any 3-track layout, vertices u and v are on different tracks, and at least one of the vertices w_i ($i = 0,\ldots,3$) is on the third track.*

Sketch of Proof. First we prove that u and v must be on different tracks. Suppose that they are both on track t. In this case at most one of the vertices w_i ($i = 0,\ldots,3$) can be on t because otherwise there would be a 4-cycle on a track and hence an overlap. Therefore at least three of the vertices w_i are on tracks different from t and since there are only 2 tracks different from t, at least two vertices w_i and w_j ($0 \le i, j \le 3$) are on a same track. Assume without loss of generality that $u < v$ and that $w_i < w_j$. Edges (u,w_j) and (v,w_i) form an X-crossing. It follows that u and v must be on two different tracks. Assume that u and v lie on tracks t_0 and t_1 respectively.

We now prove by contradiction that at least one of the vertices w_i ($i = 0,\ldots,3$) must be on the third track, t_2 say. Suppose that all the w_i ($i = 0,\ldots,3$) vertices are either on t_0 or on t_1. There are two possible cases:

1. There are three vertices w_i, w_j and w_k on the same track, t_0 say. So two of the three edges (u,w_i), (u,w_j), and (u,w_k) overlap.
2. There exist four vertices w_i, w_j, w_k, and w_h such that two of them (w_i and w_j say) are on t_0 and the other two (w_k and w_h) are on t_1. In order to avoid overlaps on t_0, u must be between w_i and w_j, i.e. $w_i < u < w_j$. Analogously $w_k < v < w_h$. Edges (u,w_k) and (v,w_i) and also edges (u,w_h) and (v,w_j) form an X-crossing.

■

Proof of Lemma 7

From the definition of block decomposition it follows that for any two consecutive blocks B_i and B_{i+1} we have $t_i < t_{i+1}$. It is also easy to see that $s_i < s_{i+1}$: if $s_{i+1} < s_i$, then the two edges (s_i,t_i) and (s_{i-1},t_{i-1}) are nested and if $s_i = s_{i+1}$ then t_i is not the vertex with the largest index connected to s_i.

Lemma 17 *Let G be a connected graph with $qn(G) = 1$. Let $\lambda = v_0, v_1, \ldots, v_{n-1}$ be the linear ordering of a 1-queue layout of G and let $B_0,\ldots,B_{k-1}$ be the block decomposition of G. We have $s_i = t_{i-1}$ if and only if s_i is a special cut vertex. Moreover $t_{i-2} \le s_i \le t_{i-1}$ ($2 \le i < k$).*

Sketch of Proof. By the definition of the block decomposition $s_i \le t_{i-1}$. If $s_i = t_{i-1}$, if follows from the definition of the block decomposition that t_{i-1} and therefore s_i is a special cut vertex. If s_i is a special cut vertex, we have $s_{i-1} < s_i \le t_{i-1}$, so from the definition of a special cut vertex it follows that $s_i = t_{i-1}$.

We now prove that $t_{i-2} \le s_i$. Suppose that $s_i < t_{i-2}$. Then $s_{i-2} < s_i < t_{i-2}$, i.e. s_i is an internal vertex of B_{i-2}. This contradicts the definition of B_{i-1} that there is no edge $e = (u,v)$ such that $u < t_{i-2}$ and $t_{i-1} < v$. Therefore we have $t_{i-2} \le s_i$. ■

Lemma 18 *Let G be a connected graph with $qn(G) = 1$ and let $B_0,\ldots,B_{k-1}$ be the block decomposition of G. The track assignment computed by Algorithm* ALPTrackLayout*() computes track numbers for vertices in B_i ($0 \le i < k$) with the following properties:*

- $track(t_i) = (track(s_i) + 2) \mod 5$
- *if s_i is a special cut vertex or $i = 0$ then for each vertex v with $s_i < v < t_i$ we have $track(v) = (track(s_i) + 1) \mod 5$*
- *if s_i is not a special cut vertex then for each vertex v with $s_i \le v < t_{i-1}$ we have $track(v) = track(s_i)$ and for each vertex v with $t_{i-1} \le v < t_i$ we have $track(v) = (track(s_i) + 1) \mod 5$.*

Sketch of Proof. The fact that $track(t_i) = (track(s_i) + 2) \mod 5$ follows from Algorithm ALPTrackLayout(). We prove the other properties by induction on the block number i. Clearly the properties hold for B_0. Assume they hold for B_{i-1}.

If s_i is a special cut vertex then $s_i = t_{i-1}$ and the result follows directly from Algorithm ALPTrackLayout(). So assume that s_i is not a special cut vertex. From Lemma 17 we know that $t_{i-2} \le s_i < t_{i-1}$. If s_{i-1} is a special cut vertex then $t_{i-2} = s_{i-1} < s_i < t_{i-1}$ and from the induction assumption we know that all vertices v with $t_{i-2} < v < t_{i-1}$ have the same track number, i.e. they have track number $track(s_i)$. If s_{i-1} is not a special cut vertex, $s_{i-1} < t_{i-2} \le s_i < t_{i-1}$, and from the inductive assumption we again conclude that all vertices v with $t_{i-2} \le s_i \le v < t_{i-1}$ have track number $track(s_i)$.

From the induction assumption we also derive that t_{i-1} has a track number that is one larger than the track number of its predecessor, so $track(t_{i-1}) = (track(s_i)+1) \mod 5$. When block B_i is considered by Algorithm ALPTRACKLAYOUT() all the internal vertices of B_i not yet assigned to a track are assigned to track $(track(s_i)+1) \mod 5$. So all vertices v with $t_{i-1} < v < t_i$ have $track(v) = (track(s_i)+1) \mod 5$. ■

Lemma 7 *Let G be a connected graph with $qn(G) = 1$. Let $\lambda = v_0, v_1, \ldots, v_{n-1}$ be the linear ordering of a 1-queue layout of G. The track assignment computed by Algorithm* ALPTRACKLAYOUT*() is such that for all edges (v_g, v_h) with $g < h$*

$$track(v_h) = (track(v_g)+1) \mod 5 \qquad or$$
$$track(v_h) = (track(v_g)+2) \mod 5.$$

Sketch of Proof. Let $B_0, \ldots, B_{k-1}$ be the block decomposition of G. We prove by induction that for each block B_i the following two invariants hold:

I1 All edges (v_g, v_h) with $s_0 \le v_g < v_h \le t_i$ are such that either $track(v_h) = (track(v_g)+1) \mod 5$ or $track(v_h) = (track(v_g)+2) \mod 5$.

I2 For all edges (v_g, v_h) with $s_0 \le v_g < t_i < v_h$ we have $track(v_g) = (track(t_i)-1) \mod 5$.

The invariant I1 holds immediately for block B_0. Also invariant I2 hold for B_0. Namely, for every edge (v_g, v_h) with $s_0 \le v_g < t_0 < v_h$ we have $s_0 < v_g$, because by definition there is no edge (s_0, v_h) with $t_0 < v_h$. The track of t_0 is 2, and the track of every edge between s_0 and t_0 is 1, thus I2 holds for B_0.

Assume the two invariants hold for B_{i-1}. We first show that I1 holds for B_i. Let (v_g, v_h) be an edge with $s_0 \le v_g < v_h \le t_i$. If $v_h \le t_{i-1}$ then I1 holds by induction. Se we may assume $t_{i-1} < v_h$. If s_i is a special cut vertex then $s_i \le v_g < v_h \le t_i$; also, since λ is a 1-queue layout of G, either $s_i = v_g$ or $v_h = t_i$. In both cases invariant I1 holds by Lemma 18. Consider the case that s_i is not a special cut vertex. Then we have $s_i < t_{i-1}$. If $s_i \le v_g$ then we have $s_i \le v_g < v_h \le t_i$ and also in this case either $s_i = v_g$ or $v_h = t_i$. It follows that invariant I1 holds by Lemma 18. If $v_g < s_i < t_{i-1}$, then from invariant I2 we derive that $track(v_g) = (track(t_{i-1})-1) \mod 5$. From Lemma 18 we know that $track(t_{i-1}) = (track(s_i)+1) \mod 5$ so $track(v_g) = track(s_i)$. From Lemma 18 we also derive that $track(v_h) = (track(s_i)+1) \mod 5$ or $track(v_h) = (track(s_i)+2) \mod 5$ so invariant I1 holds.

We prove now that invariant I2 holds for B_i. Let (v_g, v_h) be an edge with $s_0 \le v_g < t_i < v_h$. By definition there is no edge from a vertex before t_{i-1} to a vertex after t_i, therefore $t_{i-1} \le v_g$. Also, if s_i is a special cut vertex then $s_i = t_{i-1} < v_g$. From Lemma 18 we have $track(v_g) = (track(s_i)+1) \mod 5$ and $track(t_i) = (track(s_i)+2) \mod 5$, i.e. $track(v_g) = (track(t_i)-1) \mod 5$. ■

Proof of Lemma 14

Lemma 14 *Let G be a Halin graph and let G' the corresponding reduced Halin graph. Let $\Gamma(G')$ be a* 3*-track layout of G' computed by Algorithm* RHTRACKLAYOUT*() and let $e_0 = (u_0, v_0)$ and $e_1 = (u_1, v_1)$ be any pair of edges in $G - G'$. The two edges e_0 and e_1 do not have any vertex in common, and if their vertices are in the same track then either $u_0 < v_0 < u_1 < v_1$ or $v_0 < v_1 < u_0 < u_1$.*

Sketch of Proof. The edges in $G - G'$ are the long edge and the overlapping edges. The two edges e_0 and e_1 do not have a vertex in common otherwise the characteristic tree of G would have a leaf with degree two.

Also, the two vertices of e_0 are in a same track t_0 and the two vertices of e_1 are in a same track t_1, because the two vertices of each edge in $G - G'$ are the leftmost vertex and the rightmost vertex of the external path of some subtree. If $t_0 = t_1$ then either $u_0 < v_0 < u_1 < v_1$ or $v_0 < v_1 < u_0 < u_1$. Suppose as a contradiction that this is not true. At least one vertex of one edge, say e_1, must be between the two vertices of the other edge e_0. By definition, the two vertices of each overlapping edge are connected by an external path of two edges in G', and the vertices of the long edge are connected by the external path of level 0 in G'. If a vertex of e_1 is between the vertices of e_0 then the edges of the external path connecting the vertices of e_0 and the edges of the external path connecting the vertices of e_1 would form overlap. Since there is no overlap in $\Gamma(G')$ (Lemma 13) then no vertex of e_1 can be between the vertices of e_0 and vice versa. ■