Tree-Partitions of k-Trees with Applications in Graph Layout*

Vida Dujmović[1] and David R. Wood[2]

[1] School of Computer Science, McGill University, Montréal, Canada
vida@cs.mcgill.ca

[2] School of Computer Science, Carleton University, Ottawa, Canada
davidw@scs.carleton.ca

Abstract. A *tree-partition* of a graph is a partition of its vertices into 'bags' such that contracting each bag into a single vertex gives a forest. It is proved that every k-tree has a tree-partition such that each bag induces a $(k-1)$-tree, amongst other properties. Applications of this result to two well-studied models of graph layout are presented. First it is proved that graphs of bounded tree-width have bounded *queue-number*, thus resolving an open problem due to Ganley and Heath [2001] and disproving a conjecture of Pemmaraju [1992]. This result provides renewed hope for the positive resolution of a number of open problems regarding queue layouts. In a related result, it is proved that graphs of bounded tree-width have *three-dimensional straight-line grid drawings* with linear volume, which represents the largest known class of graphs with such drawings.

1 Introduction

This paper considers two models of graph layout. The first, called a *queue layout*, consists of a total order of the vertices, and a partition of the edges into *queues*, such that no two edges in the same queue are nested [11,12,15,17,20]. The dual concept of a *stack layout* (or *book embedding*), is defined similarly, except that no two edges in the same *stack* may cross. The minimum number of queues (respectively, stacks) in a queue (stack) layout of a graph is its *queue-number* (*stack-number*). Applications of queue layouts include parallel process scheduling, fault-tolerant processing, matrix computations, and sorting networks (see [15]). We prove that graphs of bounded tree-width have bounded queue-number, thus solving an open problem due to Ganley and Heath [9], who proved that stack-number is bounded by tree-width, and asked whether an analogous relationship holds for queue-number. This result has significant implications for other open problems in the field.

The second model of graph layout considered is that of a *three-dimensional* (*straight-line grid*) *drawing* [2,3,5,8,14,20]. Here vertices are positioned at grid-points in $\mathbb{Z}^3$, and edges are drawn as straight line-segments with no crossings.

* Research supported by NSERC and FCAR.

H.L. Bodlaender (Ed.): WG 2003, LNCS 2880, pp. 205–217, 2003.

While graph drawing in the plane is well-studied, there is a growing body of research in three-dimensional graph drawing. Applications include information visualisation, VLSI circuit design, and software engineering (see [5]). We focus on three-dimensional drawings with small volume, and prove that graphs of bounded tree-width have three-dimensional drawings with $\mathcal{O}(n)$ volume, which is the largest known class of graphs admitting such drawings. The best previous bound was $\mathcal{O}(n \log^2 n)$.

To prove the above results, we employ a structure called a *tree-partition* of a graph, which consists of a partition of the vertices into 'bags' such that contracting each bag to a single vertex gives a forest. In a result of independent interest, we prove that every k-tree has a tree-partition such that each bag induces a connected $(k-1)$-tree, amongst other properties. The second tool that we use is a *track layout*, which consists of a vertex-colouring and a total order of each colour class, such that between any two colour classes no two edges cross. We prove that every graph has a track layout where the number of tracks is bounded by a function of the graph's tree-width.

The remainder of the paper is organised as follows. Section 2 recalls a number of definitions and well-known results. In Section 3 we prove the above-mentioned theorem concerning tree-partitions of k-trees. In Section 4 we establish our results for track layouts. Combining these with earlier work in the companion papers [5,20], in Section 5 we prove our theorems for queue layouts and three-dimensional drawings. We discuss ramifications of our results for a number of open problems in Section 6.

2 Preliminaries

We consider undirected, simple, and finite graphs G with vertex set $V(G)$ and edge set $E(G)$. The number of vertices and maximum degree of G are respectively denoted by $n = |V(G)|$ and $\Delta(G)$. The subgraph induced by a set of vertices $A \subseteq V(G)$ is denoted by $G[A]$. A graph H is a *minor* of G if H is isomorphic to a graph obtained from a subgraph of G by contracting edges. A family of graphs closed under taking minors is *proper* if it is not the class of all graphs.

A *graph parameter* is a function α that assigns to every graph G a non-negative integer $\alpha(G)$. Let $\mathcal{G}$ be a family of graphs. By $\alpha(\mathcal{G})$ we denote the function $f : \mathbb{N} \to \mathbb{N}$, where $f(n)$ is the maximum, taken over all n-vertex graphs $G \in \mathcal{G}$, of $\alpha(G)$. We say $\mathcal{G}$ has *bounded* α if $\alpha(\mathcal{G}) \in \mathcal{O}(1)$. A graph parameter α is *bounded by* a graph parameter β, if there exists a function f such that $\alpha(G) \leq f(\beta(G))$ for every graph G.

A *k-tree* for some $k \in \mathbb{N}$ is defined recursively as follows. The empty graph is a k-tree, and the graph obtained from a k-tree by adding a new vertex adjacent to each vertex of a clique with at most k vertices is a k-tree. This definition is by Reed [16]. The following more common definition of a k-tree, which we call 'strict', was introduced by Arnborg and Proskurowski [1]. A k-clique is a *strict k-tree*, and the graph obtained from a strict k-tree by adding a new vertex adjacent to each vertex of a k-clique is a strict k-tree. Obviously the strict k-trees are a proper sub-class of the k-trees. The *tree-width* of a graph G, denoted by $\mathsf{tw}(G)$, is the minimum k such that G is a subgraph of a k-tree (which equals the minimum k such that G is a subgraph of a strict k-tree [16]). Note that k-trees can be characterised as the chordal graphs with no clique on $k+2$ vertices. Graphs with tree-width at most one are the forests. Graphs with tree-width at most two are the *series-parallel* graphs, defined as those graphs with no K_4 minor.

Let G be a graph. A total order $\sigma = (v_1, v_2, \dots, v_n)$ of $V(G)$ is called a *vertex-ordering* of G. Suppose G is connected. The *depth* of a vertex v_i in σ is the graph-theoretic distance between v_1 and v_i in G. We say σ is a *breadth-first* vertex-ordering if for all vertices $v <_\sigma w$, the depth of v in σ is no more than the depth of w in σ. Vertex-orderings, and in particular, vertex-orderings of trees will be used extensively in this paper. Consider a breadth-first vertex-ordering σ of a tree T such that vertices at depth $d \geq 1$ are ordered with respect to the ordering of vertices at depth $d-1$. In particular, if v and x are vertices at depth d with respective parents w and y at depth $d-1$ with $w <_\sigma y$ then $v <_\sigma x$. Such a vertex-ordering is called a *lexicographical* breadth-first vertex-ordering of T.

3 Tree-Partitions

Let G be a graph and let T be a tree. An element of $V(T)$ is called a *node*. Let $\{T_x \subseteq V(G) : x \in V(T)\}$ be a set of subsets of $V(G)$ indexed by the nodes of T. Each T_x is called a *bag*. The pair $(T, \{T_x : x \in V(T)\})$ is a *tree-partition* of G if:

- $\forall$ distinct nodes x and y of T, $T_x \cap T_y = \emptyset$, and
- $\forall$ edge vw of G, either
  - $\exists$ node x of T with $v \in T_x$ and $w \in T_x$ (vw is an *intra-bag* edge), or
  - $\exists$ edge xy of T with $v \in T_x$ and $w \in T_y$ (vw is an *inter-bag* edge).

The main property of tree-partitions that has been studied is the maximum size of a bag, called the *width* of the tree-partition. The minimum width over all tree-partitions of a graph G is the *tree-partition-width* of G, denoted by $\mathsf{tpw}(G)$. Ding and Oporowski [4] proved that $\mathsf{tpw}(G) \leq 24\,\mathsf{tw}(G) \cdot \max\{1, \Delta(G)\}$, and Seese [19] proved that $\mathsf{tw}(G) \leq 2\,\mathsf{tpw}(G) - 1$, for every graph G.

Theorem 1 below provides a tree-partition of a k-tree with additional features besides small width (see Figure 1). First, the subgraph induced by each bag is a connected $(k-1)$-tree. This allows us to perform induction on k. Second, in each non-root bag T_x, the vertices in the parent bag of x with a neighbour in T_x form a clique. This feature is crucial in the intended application (Theorem 2). Finally the bound on the tree-partition-width represents a constant-factor improvement over the above result by Ding and Oporowski [4] in the case of k-trees.

Theorem 1. *Let G be a k-tree with maximum degree Δ. Then G has a rooted tree-partition $(T, \{T_x : x \in V(T)\})$ such that for all nodes x of T,*

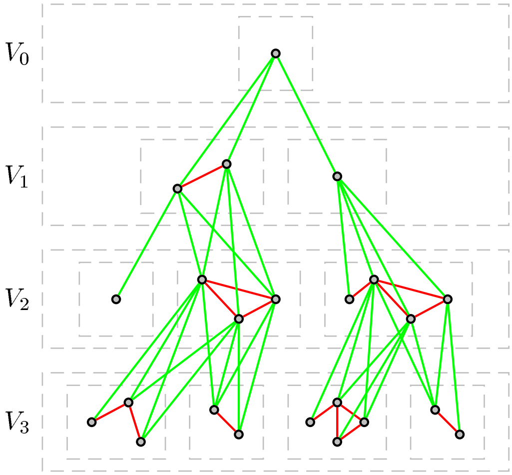

Fig. 1. Tree-partition of a 3-tree.

(a) *if x is a non-root node of T and y is the parent node of x, then the vertices in T_y with a neighbour in T_x form a clique C_x of G, and*
(b) *the induced subgraph $G[T_x]$ is a connected $(k-1)$-tree.*

Furthermore the width of $(T, \{T_x : x \in V(T)\})$ *is at most* $\max\{1, k(\Delta - 1)\}$.

Proof. We assume G is connected, since if G is not connected then a tree-partition of G that satisfies the theorem can be determined by adding a new root node with an empty bag which is adjacent to the root node of a tree-partition of each connected component of G. It is well-known[1] that for every vertex r of the k-tree G, there is a vertex-ordering $\sigma = (v_1, v_2, \dots, v_n)$ of G with $v_1 = r$, such that for all $1 \leq i \leq n$,

(i) $G^i = G[\{v_1, v_2, \dots, v_i\}]$ is connected and the vertex-ordering of G^i induced by σ is a breadth-first vertex-ordering of G^i.
(ii) the neighbours of v_i in G^i form a clique $C_i = \{v_j : v_i v_j \in E(G), j < i\}$ with $1 \leq |C_i| \leq k$ (unless $i = 1$ in which case $C_i = \emptyset$).

Let r be a vertex of minimum degree Then $\deg(r) \leq k$. Let $\sigma = (v_1, v_2, \dots, v_n)$ be a vertex-ordering of G with $v_1 = r$, and satisfying (i) and (ii). By (i), the depth of each vertex v_i in σ is the same as the depth of v_i in the vertex-ordering of G^j induced by σ, for all $j \geq i$. We therefore simply speak of *the* depth of v_i. Let V_d be the set of vertices of G at depth d.

Claim: For all $1 \leq i \leq n$, in every connected component Z of $G^i[V_d]$, the set of vertices at depth $d-1$ with a neighbour in Z form a clique of G, for all $d \geq 1$.

[1] In the language of chordal graphs, σ is a (reverse) perfect elimination vertex-ordering and can be determined, for example, by the Lex-BFS algorithm of Rose *et al.* [18].

Proof. We proceed by induction on i. The result is trivially true for $i = 1$. Suppose it is true for $i - 1$. Let d be the depth of v_i. Each vertex in C_i is at depth $d - 1$ or d. Let C'_i be the set of vertices in C_i at depth d, and let C''_i be the set of vertices in C_i at depth $d - 1$. Thus C'_i and C''_i are both cliques with $C_i = C'_i \cup C''_i$. Furthermore, if $i > 1$ then v_i must have a neighbour at depth $d - 1$, and thus $C''_i \neq \emptyset$. Let X be the vertex set of the connected component of $G^i[V_d]$ such that $v_i \in X$. By induction, for all $d' \leq d$, the claim holds for all connected components Y of $G^i[V_{d'}]$ with $Y \neq X$, since such a Y is also a connected component of $G^{i-1}[V_{d'}]$.

Case 1. $C'_i = \emptyset$: Then v_i has no neighbours in G^i at depth d; that is, $X = \{v_i\}$. Thus the set of vertices at depth $d - 1$ with a neighbour in X is precisely the clique $C_i = C''_i$.

Case 2. $C'_i \neq \emptyset$: The neighbourhood of v_i in X forms a non-empty clique (namely C'_i). Thus $X \setminus v_i$ is the vertex-set of a connected component of $G^{i-1}[V_d]$. Let Y be the set of vertices at depth $d-1$ with a neighbour in $X \setminus v_i$. By induction, Y is a clique. Since $C''_i \cup C'_i$ is a clique, $C''_i \subseteq Y$. Thus the set of vertices at depth $d - 1$ with a neighbour in X is the clique Y. □

Define a graph T and a partition $\{T_x : x \in V(T)\}$ of $V(G)$ indexed by the nodes of T as follows. There is one node x in T for every connected component of each $G[V_d]$, whose bag T_x is the vertex-set of the corresponding connected component. We say x and T_x are at *depth* d. Clearly a vertex in a depth-d bag is also at depth d. The (unique) node of T at depth zero is called the *root* node. Let two nodes x and y of T be connected by an edge if there is an edge vw of G with $v \in T_x$ and $w \in T_y$. Thus $(T, \{T_x : x \in V(T)\})$ is a 'graph-partition'. We now prove that in fact T is a tree. First observe that T is connected since G is connected. By definition, nodes of T at the same depth d are not adjacent. Moreover nodes of T can be adjacent only if their depths differ by one. Thus T has a cycle only if there is a node x in T at some depth d, such that x has at least two distinct neighbours in T at depth $d - 1$. However, by the above claim (with $i = n$), the set of vertices at depth $d - 1$ with a neighbour in T_x form a clique (called C_x), and are hence in a single bag at depth $d-1$. Thus T is a tree and $(T, \{T_x : x \in V(T)\})$ is a tree-partition of G.

We now prove that each bag T_x induces a connected $(k-1)$-tree. This is true for the root node since it only has one vertex. Suppose x is a non-root node of T at depth d. Each vertex in T_x has at least one neighbour at depth $d-1$. Thus in the vertex-ordering of T_x induced by σ, each vertex $v_i \in T_x$ has at most $k-1$ neighbours $v_j \in T_x$ with $j < i$. These neighbours induce a clique. Thus $G[T_x]$ is a $(k - 1)$-tree. By definition each $G[T_x]$ is connected.

Finally, consider the size of a bag in T. We claim that each bag contains at most $\max\{1, k(\Delta-1)\}$ vertices. The root bag has one vertex. Let x be a non-root node of T with parent node y. Suppose y is the root node. Then $T_y = \{r\}$, and thus $|T_x| \leq \deg(r) \leq k \leq k(\Delta - 1)$ assuming $\Delta \geq 2$. If $\Delta \leq 1$ then all bags have one vertex. Now assume y is a non-root node. The set of vertices in T_y with a neighbour in T_x forms the clique C_x. Let $k' = |C_x|$. Thus $k' \geq 1$, and since $C_x \subseteq T_y$ and $G[T_y]$ is a $(k - 1)$-tree, $k' \leq k$. A vertex $v \in C_x$ has $k' - 1$

neighbours in C_x and at least one neighbour in the parent bag of y. Thus v has at most $\Delta - k'$ neighbours in T_x. Hence the number of edges between C_x and T_x is at most $k'(\Delta - k')$. Every vertex in T_x is adjacent to a vertex in C_x. Thus $|T_x| \leq k'(\Delta - k') \leq k(\Delta - 1)$. This completes the proof. □

4 Track Layouts

A *colouring* of a graph G is a partition $\{V_i : i \in I\}$ of $V(G)$, where I is a set of *colours*, such that for every edge vw of G, if $v \in V_i$ and $w \in V_j$ then $i \neq j$. Each set V_i is called a *colour class*. If $<_i$ is a total order of a colour class V_i, then we call the pair $(V_i, <_i)$ a *track*. If $\{V_i : i \in I\}$ is a colouring of G, and $(V_i, <_i)$ is a track for each colour $i \in I$, then we say $\{(V_i, <_i) : i \in I\}$ is a *track assignment* of G *indexed by* I. At times it will be convenient to also refer to a colour $i \in I$ and the colour class V_i as a *track*. The precise meaning will be clear from the context. A *t-track assignment* is a track assignment with t tracks. An *X-crossing* in a track assignment consists of two edges vw and xy such that $v <_i x$ and $y <_j w$, for distinct tracks V_i and V_j. A t-track assignment with no X-crossing is called a *t-track layout*. The *track-number* of a graph G, denoted by $\mathsf{tn}(G)$, is the minimum t such that G has a t-track layout.

Dujmović *et al.* [5] first introduced track layouts[2], and proved that track-number is bounded by path-width. In particular, $\mathsf{tn}(G) \leq \mathsf{pw}(G) + 1$ for every graph G, where $\mathsf{pw}(G)$ denotes the path-width of G. In what follows we prove that track-number is bounded by tree-width. First consider the case of trees. The following result is implicit in the proof by Felsner *et al.* [8] that every outerplanar graph has a three-dimensional drawing with linear volume (see Figure 2).

Lemma 1. [8] *Every tree T has a 3-track layout.*

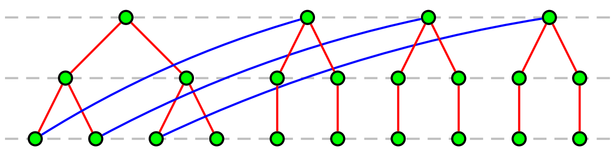

Fig. 2. A 3-track layout of a tree.

Let $\{(V_i, <_i) : i \in I\}$ be a track layout of a graph G. We say a clique C of G *covers* the set of tracks $\{i \in I : C \cap V_i \neq \emptyset\}$. Let S be a set of cliques of G. Suppose there is a total order $\preceq$ on S such that for all cliques $C_1, C_2 \in S$, if there exists a track $i \in I$, and vertices $v \in V_i \cap C_1$ and $w \in V_i \cap C_2$ with $v <_i w$, then $C_1 \prec C_2$. Then we say $\preceq$ is *nice*, and S is *nicely ordered* by the track layout. The proof of the next lemma is elementary.

---

[2] A track layout was called an ‘ordered layering with no X-crossing and no intra-layer edges’ in [5,6,20]. Similar structures are implicit in [8,11,12,17]. Note that this definition of *track-number* is unrelated to that of Gyárfás and West [10].

Lemma 2. **[6]** *Let $L \subseteq I$ be a set of tracks in a track layout $\{(V_i, <_i) : i \in I\}$ of a graph G. If S is a set of cliques, each of which covers L, then S is nicely ordered by the given track layout.*

Theorem 2. *Track-number is bounded by tree-width. In particular, every graph G with tree-width $\mathsf{tw}(G) \leq k$ has track-number $\mathsf{tn}(G) \leq t_k = 3^k \cdot 6^{(4^k-3k-1)/9}$.*

Proof. If G is not a k-tree then add edges to G to obtain a k-tree containing G as a subgraph. It is well-known that a graph with tree-width at most k is a *spanning* subgraph of a k-tree. These extra edges can be deleted once we are done. We proceed by induction on k with the following induction hypothesis:

For all $k \in \mathbb{N}$, there exist constants s_k and t_k, and sets I and S such that

1. *$|I| = t_k$ and $|S| = s_k$,*
2. *each element of S is a subset of I, and*
3. *every k-tree G has a t_k-track layout indexed by I, such that for every clique C of G, the set of tracks that C covers is in S.*

Consider the base case with $k = 0$. A 0-tree G has no edges and thus has a 1-track layout. Let $I = \{1\}$ and order $V_1 = V(G)$ arbitrarily. Thus $t_0 = 1$, $s_0 = 1$, and $S = \{\{1\}\}$ satisfy the hypothesis for every 0-tree. Now suppose the result holds for $k-1$, and G is a k-tree. Let $(T, \{T_x : x \in V(T)\})$ be a tree-partition of G described in Theorem 1, where T is rooted at r. By Theorem 1 each induced subgraph $G[T_x]$ is a $(k-1)$-tree. By induction, there are sets I and S with $|I| = t_{k-1}$ and $|S| = s_{k-1}$, such that for every node x of T, the induced subgraph $G[T_x]$ has a t_{k-1}-track layout indexed by I. For every clique C of $G[T_x]$, if C covers $L \subseteq I$ then $L \in S$. Assume $I = \{1, 2, \dots, t_{k-1}\}$ and $S = \{S_1, S_2, \dots, S_{s_{k-1}}\}$. By Theorem 1, for each non-root node x of T, if p is the parent node of x, then the set of vertices in T_p with a neighbour in T_x form a clique C_x. Let $\alpha(x) = i$ where C_x covers S_i. Let $\alpha(r) = 1$.

To construct a track layout of G we first construct a track layout of T indexed by $\{(d, i) : d \geq 0, 1 \leq i \leq s_{k-1}\}$, where the track $L_{d,i}$ consists of nodes x of T at depth d with $\alpha(x) = i$. Here the *depth* of a node x is the distance in T from the root node r to x. We order the nodes of T within the tracks by increasing depth. There is only one node at depth $d = 0$. Suppose we have determined the orders of the nodes up to depth $d-1$ for some $d \geq 1$. Let $i \in \{1, 2, \dots, s_{k-1}\}$. The nodes in $L_{d,i}$ are ordered primarily with respect to the relative positions of their parent nodes (at depth $d-1$). More precisely, let $\rho(x)$ denote the parent node of each $x \in L_{d,i}$. For all nodes x and y in $L_{d,i}$, if $\rho(x)$ and $\rho(y)$ are in the same track and $\rho(x) < \rho(y)$ in that track, then $x < y$ in $L_{d,i}$. For x and y with $\rho(x)$ and $\rho(y)$ on distinct tracks, the relative order of x and y is not important. It remains to specify the order of nodes in $L_{d,i}$ with a common parent. Suppose P is a set of nodes in $L_{d,i}$ with a common parent node p. By construction, for every node $x \in P$, the parent clique C_x covers S_i in the track layout of $G[T_p]$. By Lemma 2 the cliques $\{C_x : x \in P\}$ are nicely ordered by the track layout of $G[T_p]$. Let the order of P in track $L_{d,i}$ be specified by a nice ordering of $\{C_x : x \in P\}$, as

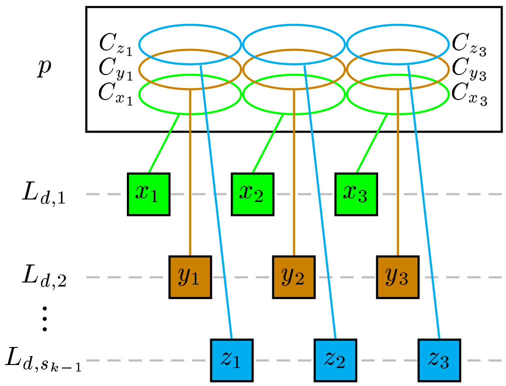

Fig. 3. Nodes with a common parent p.

illustrated in Figure 3. This construction defines a partial order on the nodes in track $L_{d,i}$ that can be arbitrarily extended to a total order. Hence we have a track assignment of T. Since the nodes in each track are ordered primarily with respect to the relative positions of their parent nodes in the previous tracks, there is no X-crossing, and hence we have a track layout of T.

To construct a track assignment of G from the track layout of T, replace each track $L_{d,i}$ by t_{k-1} 'sub-tracks', and for each node x of T, insert the track layout of $G[T_x]$ in place of x on the sub-tracks corresponding to the track containing x. More formally, the track assignment of G is indexed by $\{(d,i,j) : d \geq 0, 1 \leq i \leq s_{k-1}, 1 \leq j \leq t_{k-1}\}$. Each track $V_{d,i,j}$ consists of those vertices v of G such that, if T_x is the bag containing v, then x is at depth d in T, $\alpha(x) = i$, and v is on track j in the track layout of $G[T_x]$. If x and y are distinct nodes of T with $x < y$ in $L_{d,i}$, then $v < w$ in $V_{d,i,j}$, for all vertices $v \in T_x$ and $w \in T_y$ on track j. If v and w are vertices of G on track j in bag T_x at depth d, then the relative order of v and w in $V_{d,\alpha(x),j}$ is the same as in the track layout of $G[T_x]$.

Clearly adjacent vertices of G are in distinct tracks. Thus we have defined a track assignment of G. We claim that there is no X-crossing. Clearly an intra-bag edge of G is not in an X-crossing with an edge not in the same bag. By induction, there is no X-crossing between intra-bag edges in a common bag. Since there is no X-crossing in the track layout of T, inter-bag edges of G which are mapped to edges of T without a common parent node, are not involved in an X-crossing. Consider a parent node p in T. For each child node x of p, the vertices in T_p adjacent to a vertex in T_x forms the clique C_x. Thus there is no X-crossing between a pair of edges both from C_x to T_x, since the vertices of C_x are on distinct tracks. Consider two child nodes x and y of p. For there to be an X-crossing between an edge from T_p to T_x and an edge from T_p to T_y, the nodes x and y must be on the same track in the track layout of T. Suppose $x < y$ in this track. By construction, C_x and C_y cover the same set of tracks, and $C_x \preceq C_y$ in the corresponding nice ordering. Thus for any track containing vertices $v \in C_x$ and $w \in C_y$, $v \leq w$ in that track. Since all the vertices in T_x are to the left of the vertices in T_y (on a common track), there is no X-crossing between an edge from T_p to T_x and an edge from T_p to T_y. Therefore there is no X-crossing, and hence we have a track layout of G.

We now 'wrap' the track layout of G. Define a track assignment of G indexed by $\{(d', i, j) : d' \in \{0,1,2\}, 1 \leq i \leq s_{k-1}, 1 \leq j \leq t_{k-1}\}$, where each track $W_{d',i,j} = \bigcup\{V_{d,i,j} : d \equiv d' \pmod 3\}$. If $v \in V_{d,i,j}$ and $w \in V_{d+3,i,j}$ then $v < w$ in the order of $W_{d',i,j}$ (where $d' = d \bmod 3$). The order of each $V_{d,i,j}$ is preserved in $W_{d',i,j}$. The tracks $\{W_{d',i,j} : d' \in \{0,1,2\}, 1 \leq i \leq s_{k-1}, 1 \leq j \leq t_{k-1}\}$ forms a track assignment of G. For every edge vw of G, the depths of the bags in T containing v and w differ by at most one. Thus in the wrapped track assignment of G, adjacent vertices remain on distinct tracks, and there is no X-crossing. The number of tracks is $3 \cdot s_{k-1} \cdot t_{k-1}$. Every clique C of G is either contained in a single bag of the tree-partition or is contained in two adjacent bags. Let $S' = \{\{(d', i, h) : h \in S_j\} : d' \in \{0,1,2\}, 1 \leq i, j \leq s_{k-1}\}$. For every clique C of G contained in a single bag, the set of tracks containing C is in S'. Let $S'' = \{\{(d', i, h) : h \in S_j\} \cup \{((d'+1) \bmod 3, p, h) : h \in S_q\} : d' \in \{0,1,2\}, 1 \leq i, j, p, q \leq s_{k-1}\}$. For every clique C of G contained in two bags, the set of tracks containing C is in S'. Observe that $S' \cup S''$ is independent of G. Hence $S' \cup S''$ satisfies the hypothesis for k. Now $|S'| = 3s_{k-1}^2$ and $|S''| = 3s_{k-1}^4$, and thus $|S' \cup S''| = 3s_{k-1}^2(s_{k-1}^2 + 1)$. Therefore any solution to the recurrences $\{s_0 \geq 1,\ t_0 \geq 1,\ s_k \geq 3s_{k-1}^2(s_{k-1}^2 + 1),\ t_k \geq 3s_{k-1} \cdot t_{k-1}\}$ satisfies the theorem. It is easily verified that $s_k = 6^{(4^k-1)/3}$ and $t_k = 3^k \cdot 6^{(4^k-3k-1)/9}$ is such a solution. □

A number of refinements to the proof of Theorem 2 that result in improved bounds are possible [6]. For example, in the case of $\mathsf{tw}(G) = 2$, we prove that $\mathsf{tn}(G) \leq 18$, whereas Theorem 2 proves that $\mathsf{tn}(G) \leq 54$. One such refinement uses strict k-trees. From an algorithmic point of view, the disadvantage of using strict k-trees is that at each recursive step, extra edges must be added to enlarge the graph into a strict k-tree, whereas when using (non-strict) k-trees, extra edges need only be added at the beginning of the algorithm.

If maximum degree as well as tree-width is bounded then the dependence on the tree-width in our track-number bound can be substantially reduced.

Theorem 3. *Every graph G with maximum degree $\Delta(G)$, tree-width $\mathsf{tw}(G)$, and tree-partition-width $\mathsf{tpw}(G)$, has track-number $\mathsf{tn}(G) \leq 3\,\mathsf{tpw}(G) \leq 72\,\mathsf{tw}(G) \cdot \max\{1, \Delta(G)\}$.*

Proof. Let $(T, \{T_x : x \in V(T)\})$ be a tree-partition of G with width $\mathsf{tpw}(G)$. By Lemma 1, T has a 3-track layout. Replace each track by $\mathsf{tpw}(G)$ 'sub-tracks', and for each node x in T, place the vertices in T_x on the sub-tracks replacing the track containing x, with at most one vertex in T_x on a single track. The total order of each sub-track preserves the total order in each track of the track-layout of T. There is no X-crossing, since in the track layout of T, adjacent nodes are on distinct tracks and there is no X-crossing. Thus we have a track layout of G with $3\,\mathsf{tpw}(G) \leq 72\,\mathsf{tw}(G) \cdot \max\{1, \Delta(G)\}$ tracks [4]. □

5 Queue Layouts and 3D Graph Drawings

A *queue layout* of a graph G consists of a vertex-ordering σ of G, and a partition of $E(G)$ into *queues*, such that no two edges in the same queue are *nested* with respect to σ. That is, there are no edges vw and xy in a single queue with $v <_\sigma x <_\sigma y <_\sigma w$. A similar concept is that of a *stack layout* (or *book embedding*), which consists of a vertex-ordering σ of G, and a partition of $E(G)$ into *stacks* (or *pages*) such that there are no edges vw and xy in a single stack with $v <_\sigma x <_\sigma w <_\sigma y$. The minimum number of queues (respectively, stacks) in a queue (stack) layout of G is called the *queue-number* (*stack-number* or *page-number*) of G, and is denoted by $\mathsf{qn}(G)$ ($\mathsf{sn}(G)$). Ganley and Heath [9] proved that stack-number is bounded by tree-width, and asked whether queue-number is also bounded by tree-width? The bound of $\mathsf{sn}(G) \leq \mathsf{tw}(G) + 1$ by Ganley and Heath [9] has recently been improved to $\mathsf{sn}(G) \leq \mathsf{tw}(G)$ by Lin and Li [13].

A 1-tree has queue-number at most one, since in a lexicographical breadth-first vertex-ordering of a tree no two edges are nested [12]. Rengarajan and Veni Madhavan [17] proved that 2-trees have queue-number at most three. Wood [20] proved that queue-number is bounded by path-width and tree-partition-width. In particular, $\mathsf{qn}(G) \leq \mathsf{pw}(G)$ and $\mathsf{qn}(G) \leq \frac{3}{2}\mathsf{tpw}(G)$ for every graph G. Hence $\mathsf{qn}(G) \leq 36\,\mathsf{tw}(G) \cdot \max\{1, \Delta(G)\}$ by the result of Ding and Oporowski [4]. Wood [20] also proved that $\mathsf{qn}(G) \leq \mathsf{tn}(G) - 1$ for every graph G. Thus Theorem 2 implies the following result, which answers the above question of Ganley and Heath [9] in the affirmative. Further consequences are discussed in Section 6.

Theorem 4. *Queue-number is bounded by tree-width. In particular, every graph G with tree-width $\mathsf{tw}(G) \leq k$ has queue-number $\mathsf{qn}(G) < 3^k \cdot 6^{(4^k-3k-1)/9}$.*

A *three-dimensional straight-line grid drawing* of a graph, henceforth called a *3D drawing*, represents the vertices by distinct points in $\mathbb{Z}^3$ (called *grid-points*), and represents each edge as a line-segment between its end-vertices, such that edges only intersect at common end-vertices, and an edge only intersects a vertex that is an end-vertex of that edge. In contrast to the case in the plane, it is well known that every graph has a 3D drawing. We therefore are interested in optimising certain measures of the aesthetic quality of a drawing. If a 3D drawing is contained in an axis-aligned box with side lengths $X - 1$, $Y - 1$ and $Z - 1$, then we speak of a $X \times Y \times Z$ drawing with *volume* $X \cdot Y \cdot Z$. We study 3D drawings with small volume.

Cohen *et al.* [2] proved that every graph has a 3D drawing with $\mathcal{O}(n^3)$ volume, and this bound is asymptotically tight for K_n. It is therefore of interest to identify fixed graph parameters that allow for 3D drawings with $o(n^3)$ volume. Pach *et al.* [14] proved that graphs of bounded chromatic number have 3D drawings with $\mathcal{O}(n^2)$ volume, and that this bound is asymptotically optimal for $K_{n,n}$. The first non-trivial $\mathcal{O}(n)$ volume bound was established by Felsner *et al.* [8] for outerplanar graphs. Dujmović *et al.* [5,20] proved that track layouts, queue layouts, and 3D drawings with small volume are inherently related.

Theorem 5. [5,20] *Every n-vertex graph G has a $\mathcal{O}(\mathsf{tn}(G)) \times \mathcal{O}(\mathsf{tn}(G)) \times \mathcal{O}(n)$ drawing. Let $\mathcal{F}(n)$ be a family of functions closed under multiplication, such as $\mathcal{O}(1)$ or $\mathcal{O}(\text{polylog}\, n)$. Then for any graph family $\mathcal{G}$, every graph $G \in \mathcal{G}$ has a $\mathcal{F}(n) \times \mathcal{F}(n) \times \mathcal{O}(n)$ drawing if and only if the track-number $\mathsf{tn}(\mathcal{G}) \in \mathcal{F}(n)$. Moreover, if $\mathcal{G}$ is proper minor-closed then $\mathcal{G}$ has track-number $\mathsf{tn}(\mathcal{G}) \in \mathcal{F}(n)$ if and only if $\mathcal{G}$ has queue-number $\mathsf{qn}(\mathcal{G}) \in \mathcal{F}(n)$.*

Applying Theorem 5, Dujmović *et al.* [5] proved that every graph G has a 3D drawing with $\mathcal{O}(\mathsf{pw}(G)^2 \cdot n)$ volume, which is $\mathcal{O}(n \log^2 n)$ for graphs of bounded tree-width. Using the result of Rengarajan and Veni Madhavan [17] discussed in Section 5, Wood [20] proved that series-parallel graphs have 3D drawings with $\mathcal{O}(n)$ volume, but with a constant of at least 10^{16}. For particular sub-classes of series-parallel graphs, improved constants have been obtained [3].

Wood [20] proved that graphs of bounded tree-partition-width have 3D drawings with $\mathcal{O}(n)$ volume, although the actual volume bound is approximately $\mathcal{O}(\mathsf{tw}(G)^4(\mathsf{tw}(G)^2\,\mathsf{tpw}(G))^{\mathsf{tw}(G)^2} \cdot n)$. Theorems 3 and 5 together prove the following result, which represents a substantial improvement in the dependence on $\mathsf{tpw}(G)$ compared with the above-mentioned result.

Theorem 6. *Every n-vertex graph G with bounded tree-partition-width, which includes graph of bounded tree-width and bounded degree, has a 3D drawing with $\mathcal{O}(n)$ volume. In particular, the drawing is $\mathcal{O}(\mathsf{tpw}(G)) \times \mathcal{O}(\mathsf{tpw}(G)) \times \mathcal{O}(n)$, which is $\mathcal{O}(\mathsf{tw}(G)\, \Delta(G)) \times \mathcal{O}(\mathsf{tw}(G)\, \Delta(G)) \times \mathcal{O}(n)$.*

Theorems 2 and 5 together prove our main result of this section.

Theorem 7. *Every n-vertex graph G with bounded tree-width has a 3D drawing with $\mathcal{O}(n)$ volume. In particular, the drawing is $\mathcal{O}(6^{4^{\mathsf{tw}(G)}}) \times \mathcal{O}(6^{4^{\mathsf{tw}(G)}}) \times \mathcal{O}(n)$.*

As well as providing many new classes of graphs that admit 3D drawings with $\mathcal{O}(n)$ volume, Theorem 7 dramatically improves the constant in the bound for series-parallel graphs. As mentioned in Section 4, such graphs have 18-track layouts. It follows that every series-parallel graph has a $36 \times 37 \times 37 \lceil \frac{n}{18} \rceil$ drawing.

6 Open Problems

Consider the following open problems: (1) Do planar graphs have bounded queue-number? (2) Is queue-number bounded by stack-number? Since planar graphs have bounded stack-number, the second question is more general than the first. Heath *et al.* [11] conjectured that both of these questions have an affirmative answer. More recently however, Pemmaraju [15] conjectured that the 'stellated K_3', a planar 3-tree, has $\Theta(\log n)$ queue-number, and provided evidence to support this conjecture (also see [9]). This suggested that the answers to the above questions were both negative. In particular, Pemmaraju [15] and Heath [private communication, 2002] conjectured that planar graphs have $\mathcal{O}(\log n)$ queue-number. However, Theorem 4 provides a queue-layout of *any* 3-tree, and thus the stellated K_3, with $\mathcal{O}(1)$ queues. Hence our result disproves the first conjecture of Pemmaraju [15] mentioned above, and renews hope in an affirmative answer to the above open problems. By Theorem 5, question (1) is equivalent to the question of whether planar graphs have bounded track-number, which was asked by H. de Fraysseix [private communication, 2000] in the context of graph drawing. If planar graphs have bounded track-number then such graphs would also admit 3D drawings with $\mathcal{O}(n)$ volume, which is an open problem due to Felsner *et al.* [8]. The authors recently proved that planar graphs and graphs of bounded degree have 3D drawings with $\mathcal{O}(n^{3/2})$ volume [7].

References

1. S. Arnborg and A. Proskurowski. Linear time algorithms for NP-hard problems restricted to partial k-trees. *Discrete Appl. Math.* **23**(1):11–24, 1989.
2. R. F. Cohen, P. Eades, T. Lin, and F. Ruskey. Three-dimensional graph drawing. *Algorithmica* **17**(2):199–208, 1996.
3. E. Di Giacomo, G. Liotta, and S. Wismath. Drawing series-parallel graphs on a box. In *Proc. 14th Canadian Conf. on Computational Geometry* (CCCG '02), pages 149–153. The Univ. of Lethbridge, Canada, 2002.
4. G. Ding and B. Oporowski. Some results on tree decomposition of graphs. *J. Graph Theory* **20**(4):481–499, 1995.
5. V. Dujmović, P. Morin, and D. R. Wood. Path-width and three-dimensional straight-line grid drawings of graphs. In *Proc. 10th Symp. on Graph Drawing* (GD '02), LNCS **2628**:42–53, Springer, 2002.
6. V. Dujmović and D. R. Wood. Tree-partitions of k-trees with applications in graph layout. Technical Report TR-02-03, School of Computer Science, Carleton Univ., Ottawa, Canada, 2002.
7. V. Dujmović and D. R. Wood. Three-Dimensional Grid Drawings with Sub-Quadratic Volume. In *Proc. 11th Symp. on Graph Drawing* (GD '03), LNCS, Springer, to appear. Also in: Towards a Theory of Geometric Graphs, *Contemporary Mathematics*, Amer. Math. Soc., to appear.
8. S. Felsner, G. Liotta, and S. Wismath. Straight-line drawings on restricted integer grids in two and three dimensions. In *Proc. 9th Symp. on Graph Drawing* (GD '01), LNCS **2265**:328–342, Springer, 2002.
9. J. L. Ganley and L. S. Heath. The pagenumber of k-trees is $O(k)$. *Discrete Appl. Math.* **109**(3):215–221, 2001.
10. A. Gyárfás and D. West. Multitrack interval graphs. In *26th Southeastern Conf. on Combinat., Graph Theory and Comput., Congr. Numer.* **109**:109–116, 1995.
11. 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**:398–412, 1992.
12. L. S. Heath and A. L. Rosenberg. Laying out graphs using queues. *SIAM J. Comput.* **21**(5):927–958, 1992.
13. Y. Lin, and X. Li. Pagenumber and treewidth. *Disc. Applied Math.*, to appear.
14. J. Pach, T. Thiele, and G. Tóth. Three-dimensional grid drawings of graphs. In *Proc. 5th Symp. on Graph Drawing* (GD '97), LNCS **1353**:47–51, Springer, 1997. Also in: Advances in discrete and computational geometry, *Contemporary Mathematics* **223**:251–255, Amer. Math. Soc., 1999.
15. S. V. Pemmaraju. *Exploring the Powers of Stacks and Queues via Graph Layouts.* PhD thesis, Virginia Polytechnic Institute and State Univ., Virginia, U.S.A., 1992.

16. B. A. Reed. Algorithmic aspects of tree width. In *Recent advances in algorithms and combinatorics*, pages 85–107, Springer, 2003.
17. S. Rengarajan and C. E. Veni Madhavan. Stack and queue number of 2-trees. In *Proc. 1st Conf. on Computing and Combinatorics* (COCOON '95), LNCS **959**:203–212, Springer, 1995.
18. D. J. Rose, R. E. Tarjan, and G. S. Leuker. Algorithmic aspects of vertex elimination on graphs. *SIAM J. Comput.* **5**:266–283, 1976.
19. D. Seese. Tree-partite graphs and the complexity of algorithms. In *Proc. Conf. Fundamentals of Computation Theory*, LNCS **199**:412–421, Springer, 1985.
20. D. R. Wood. Queue layouts, tree-width, and three-dimensional graph drawing. In *Proc. 22nd Foundations of Software Technology and Theoretical Computer Science* (FST TCS '02), LNCS **2556**:348–359, Springer, 2002.