Pathwidth and Layered Drawings of Trees

Matthew Suderman

Technical Report SOCS-02.8
October 2002

School of Computer Science
McGill University

Pathwidth and Layered Drawings of Trees

Matthew Suderman*
msuder@cs.mcgill.ca
School of Computer Science, McGill University
Montréal, Québec, Canada

December 10, 2002

Abstract

An *h-layer drawing* of a graph G is a planar drawing of G in which each vertex is placed on one of h parallel lines and each edge is drawn as a straight line between its end-vertices. In such a drawing, we say that an edge is *proper* if its endpoints lie on adjacent layers, *flat* if they lie on the same layer and *long* otherwise. Thus, a *proper* h-layer drawing contains only proper edges, a *short* h-layer drawing contains no long edges, an *upright* h-layer drawing contains no flat edges, and an *unconstrained* h-layer drawing contains any type of edge.

We prove optimal upper and lower bounds for each type of layered drawing (proper, short, upright, unconstrained) and give linear-time algorithms for obtaining drawings matching each upper bound. We note that the optimality of the upper bound for unconstrained layered drawings contradicts Proposition 1 of [8], and the optimality of the upper bound for short layered drawings contradicts Theorem 2 also of [8].

1 Introduction

An *h-layer drawing* of a graph G is a planar drawing of G in which each vertex is placed on one of h parallel lines and each edge is drawn as a straight line between its end-vertices. In such a drawing, we say that an edge is *proper* if its endpoints lie on adjacent layers, *flat* if they lie on the same layer and *long* otherwise. Thus, a *proper* h-layer drawing contains only proper edges, a *short* h-layer drawing contains no long edges, an *upright* h-layer drawing contains no flat edges, and an *unconstrained* h-layer drawing contains any type of edge. Layered graph drawings [18, 2, 17] have applications in visualization [1, 12], DNA mapping [19], and VLSI layout [13]. See [14] for a recent survey.

In this paper we show that every short layered drawing of a tree T with pathwidth $h \geq 2$ occupies between h and $2h-1$ layers. Furthermore, we prove that these bounds are optimal.

*This research supported by NSERC and FQRNT (formerly FCAR).

We similarly prove optimal upper and lower bounds for proper layered drawings (between h and $3h-2$ layers for $h \geq 2$), upright layered drawings (between h and $\lceil 3h/2 \rceil$ layers for $h \geq 1$) and unconstrained layered drawings (between $h-1$ and $\lceil 3h/2 \rceil$ layers for $h \geq 1$). Finally, we give linear-time algorithms for obtaining short, proper, upright and unconstrained drawings matching the upper bounds.

These results contradict Proposition 1 and Theorem 2 of [8]. In particular, Proposition 1 states that a tree with pathwidth h can be drawn on an $n \times h$ grid. However, in this paper we describe trees that cannot be drawn on less than $3h/2$ layers. Similarly, Theorem 2 implies that such a tree has a short $(h+1)$-layer drawing. However, we describe trees whose short layered drawings occupy at least $2h-1$ layers.

2 Preliminaries

In an h-layer drawing of a graph G, we number the layers consecutively from 1 to h, with layer 1 as the top layer and layer h as the bottom layer. We use, for each vertex $v \in V(G)$, $\mathrm{X}(v)$ to denote the x-coordinate and $\mathrm{Y}(v)$ to denote the y-coordinate of v. The following simple lemma states one of the key observations that we use to establish lower bounds on the number of layers used in layered tree drawings.

Lemma 1 *In any unconstrained h-layer drawing of a tree T with a vertex v, the drawings of at most two components of $T \setminus v$ occupy h layers.*

Proof. Assume the contradiction, that $T \setminus v$ contains at least three components T_1, T_2 and T_3 whose drawings each occupy h layers. Each T_i occupies all h layers so each has a vertex v_i on the top layer. Assume without loss of generality that $\mathrm{X}(v_1) \leq \mathrm{X}(v_2) \leq \mathrm{X}(v_3)$. However, T_2 has another vertex v_2' on the bottom layer so the drawing is not planar: an edge in the path from v_1 to v_3 in $T \setminus T_2$ crosses an edge in the path from v_2 to v_2' in T_2. □

From Lemma 1, we obtain the following result about layered drawings of complete ternary trees, which was already proven in [8]:

Corollary 2 *Every unconstrained layered drawing of a complete ternary tree of depth $d \geq 0$ occupies at least $d+1$ layers.*

Proof. The proof is by induction on the depth d of the tree beginning at depth $d = 0$ where the tree consists of a single vertex. □

We also obtain the following bounds for upright layered drawings of nearly complete ternary trees. A *nearly complete ternary tree* of depth d is obtained from a complete ternary tree of depth d by removing exactly two children from each vertex at depth $d-1$.

Corollary 3 *Every upright layered drawing of a nearly complete ternary tree of depth $d \geq 0$ occupies at least $d+1$ layers.*

Proof. The proof is identical to the proof of Corollary 2 above. □

Nearly all of the remaining results depend on the pathwidth of the given tree. A *path decomposition* B of a graph G is a sequence $B_1,\ B_2,\ \ldots,\ B_p$ of subsets of $V(G)$ that satisfies the following three properties:

1. $\bigcup_{1\le i\le p} B_i = V(G)$;
2. for every edge $(u,v)\in E(G)$, there is a subset B_i such that both $u,v\in B_i$; and
3. for all $1\le i<j<k\le p$, $B_i\cap B_k\subseteq B_j$.

The *width* of B is $\max\{|B_i| \mid 1\le i\le p\}-1$. The *pathwidth* of a graph G, denoted $\mathrm{pw}(G)$, is the minimum width of a path decomposition of G. Linear algorithms for computing the pathwidth of trees are described in [15, 7, 16].

The next two results about trees and pathwidth are given in [15, 7]:

Lemma 4 *[15, 7] A tree T has pathwidth at most h if and only if for all vertices v in T at most two components of $T\setminus v$ have pathwidth h and the remainder have pathwidth at most $h-1$.*

As defined in [7], we say that a vertex v is h-critical in a rooted tree T if exactly two subtrees rooted at children of v have pathwidth h and the remainder have pathwidth at most $h-1$.

Lemma 5 *[7] Let T be a tree rooted at r. If at most two subtrees rooted at children of r have pathwidth h, neither has an h-critical vertex, and every other subtree rooted at a child of r has pathwidth at most $h-1$, then $\mathrm{pw}(T)\le h$.*

Finally, the following lemma is given in [7, 16]:

Lemma 6 *[7, 16] A tree T has at most one $\mathrm{pw}(T)$-critical vertex.*

In the next section we prove optimal upper and lower bounds on the number of layers required by short layered drawings of trees. We follow that with optimal upper and lower bounds for proper, upright and unconstrained layered drawings in Sections 4 and 5, and finally, in Section 6, we give linear-time algorithms for obtaining short, proper, upright and unconstrained drawings matching the upper bounds.

We prove our upper bounds by constructing drawings of trees. Similar to Felsner *et al.* [8], the drawings are constructed in two steps: we draw one or more paths in the tree and then recursively draw the remaining components next to the previously drawn paths. The most important of these paths is the *main* path. A *main* path P of a tree T is a path such that the pathwidth of $T\setminus P$ is at most $\mathrm{pw}(T)-1$.

Lemma 7 *Every tree has at least one main path.*

PROOF. Consider a path decomposition $B=B_1,\ B_2,\ \ldots,\ B_p$ of T of minimum width. Let v_1 be a vertex in B_1, v_p a vertex in B_p, and P the path between v_1 and v_p. By definition, each B_i contains at least one vertex in P so, if we remove the vertices of P from each B_i, then the result is a path decomposition of $T\setminus P$ with width at most $\mathrm{pw}(T)-1$. □

The remaining components must be drawn so that we can insert the edges connecting the components to the previously drawn paths without creating crossings. As a result, if Γ is a drawing of a component and vertex v in the component is adjacent to a previously drawn path vertex, then v lies on the top or bottom layer of Γ. We say that v is *exposed* in Γ. In general, a vertex $v \in T$ is *exposed* in a layered drawing of a tree T if v lies on the top or bottom layer of the drawing.

The next result illustrates how to obtain proper 2-layer drawings of certain trees given one of their main paths.

Lemma 8 *For every tree T with $\mathrm{pw}(T) \leq 1$, there exists a proper 2-layer drawing.*

PROOF. By Lemma 7, T has a main path P, and $T \setminus P$ consists of vertices with degree zero. We draw T by first drawing P on both layers and then inserting each vertex v in $T \setminus P$ adjacent to a vertex $w \in P$ on the layer opposite w. $\square$

3 Short Layered Drawings

Using a similar though slightly more complicated drawing algorithm than in Lemma 8, we obtain an upper bound for short layered drawings, first proved in [5]:

Lemma 9 *Every tree T with $\mathrm{pw}(T) \geq 2$ has a short $(2\mathrm{pw}(T) - 1)$-layer drawing.*

PROOF. We obtain a short $(2\mathrm{pw}(T) - 1)$-layer drawing by first drawing a main path P of T on the top layer. Each component T' in $T \setminus P$ has pathwidth at most $\mathrm{pw}(T) - 1$. Let v be the vertex in T' adjacent to a vertex w in P. We insert a drawing of each T' onto the $(2\mathrm{pw}(T) - 2)$ layers below w and then draw the missing edge (v, w) as a straight line between v and w. To avoid edge-crossings, we draw T' so that v lies on the layer immediately below w; that is, we draw T' so that v is exposed. It remains to prove, then, that such drawings of T' exist. In other words, we must prove that for any tree T with $\mathrm{pw}(T) \geq 1$ and vertex $v \in T$, there exists a short $2\mathrm{pw}(T)$-layer drawing of T in which v is exposed.

The proof is by induction on the pathwidth of T. When $\mathrm{pw}(T) = 1$, there is a short 2-layer drawing of T by Lemma 8. Since there are only two layers, every vertex including v is exposed. Suppose that $\mathrm{pw}(T) \geq 2$ and let $P = v_1 v_2 \dots v_n$ be a main path in T and R the path between v and a vertex v_i in P. We begin drawing T on $2\mathrm{pw}(T)$ layers by first drawing the path $R v_i v_{i-1} \dots v_1$ on the top layer and then the path $v_{i+1} v_{i+2} \dots v_n$ on the second layer below edge (v_i, v_{i-1}). Each connected component T' of $(T \setminus P) \setminus R$ has pathwidth at most $\mathrm{pw}(T) - 1$ so, by induction, there exists a short $(2\mathrm{pw}(T) - 2)$-layer drawing of T' in which vertex $v' \in T'$ adjacent to vertex w in $P \cup R$ is exposed. We recursively construct and then insert this drawing of T' onto the layers below w. The final drawing is illustrated in Figure 1. $\square$

This upper bound is optimal for a set of rooted trees that, when drawn on a minimum number of layers, the root is not exposed. To achieve this property, we require the following result:

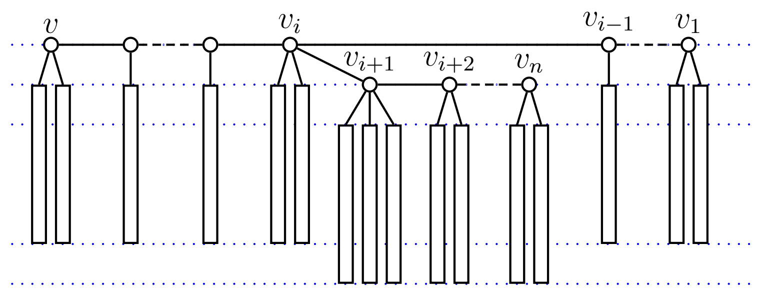

Figure 1: A short $(2\text{pw}(T)-1)$-layer drawing T in which vertex v in T is exposed.

Lemma 10 *Let $h \geq 1$ and T be a tree rooted at a vertex r with $n \geq 0$ children. Suppose that each subtree rooted at a child c of r has the property that every short layered drawing occupies at least $h-1$ layers and at least h layers if c is exposed. If $n \geq (i+3)(i+2)+1$ for some $0 \leq i < h$ then, in any short $(h+i)$-layer drawing of T, r is on one of the top or bottom i layers.*

PROOF. Assume by way of contradiction that r does not lie on the top or bottom i layers; that is, r lies on layer j, $i < j < h+1$. If the drawing of a subtree rooted at a child c of r occupies exactly $l = h-1$ layers then c is not exposed in that drawing so r lies on one of those l layers. If instead the subtree occupies $l \geq h$ layers then these l layers include layers $i+1,\ i+2,\ \ldots,\ h$, one which is occupied by r. By Lemma 1, then, the drawings of at most two subtrees occupy the same set of layers. There are $h+i-l+1$ ways to choose $l \geq h-1$ consecutive layers from $h+i$ total layers so r can have at most $2\Sigma_{l=h-1}^{h+i} h+i-l+1 = 2[(i+2)+(i+1)+\ldots+1] = (i+2)(i+3)$ children. $\square$

We describe the set of trees recursively. For $k=1$, we define $S^k = S^1$ to be the complete ternary tree of height one. For $k \geq 2$, S^k consists of a root v with one child x that has two children u and w. In addition, we make use of Lemma 10 when $i=1$ by giving u, w and x each $n=13$ children. Each child is a root of a subtree isomorphic to S^{k-1}. See Figure 2.

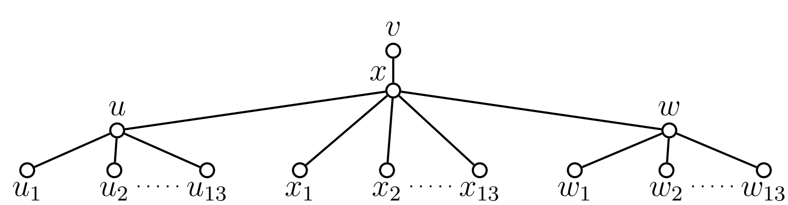

Figure 2: Tree S^k for $k \geq 2$. Each u_i, w_i and x_i is the root of a subtree isomorphic to S^{k-1}.

Lemma 11 *For $k \geq 1$, every short layered drawing of S^k occupies at least $2k-1$ layers and at least $2k$ layers if its root v is exposed.*

PROOF. The proof is by induction on k. For $k = 1$, $S^k = S^1$ is a complete ternary tree of depth 1 so by Corollary 2 every drawing occupies at least $2k = 2$ layers.

Assume that $k \geq 2$. By induction, every short layered drawing of S^{k-1} occupies at least $2k-3$ layers and $2k-2$ layers if its root is exposed. By Lemma 10, then, a drawing of subtree S^k_u, S^k_w or $(S^k_x \setminus S^k_u) \setminus S^k_w$ occupies at least $2k-1$ layers and, if exactly $2k-1$ layers, then the root (u, w or x) is exposed. If v is exposed in a $(2k-1)$-layer drawing of S^k then either u, w or x is not exposed in the drawing of its corresponding subtree. However, in that case, the corresponding subtree occupies $2k$ layers; therefore, the drawing of S^k occupies at least $2k$ layers. □

We obtain an upper bound on the pathwidth of each S^k.

Lemma 12 *For* $k \geq 1$, $\mathrm{pw}(S^k) \leq k$.

PROOF. The proof is by induction on k. For $k = 1$, the pathwidth of $S^k = S^1$ is 1. For $k \geq 2$, there are, by induction, path decompositions for each $S^k_{u_i}$, $S^k_{w_i}$ and $S^k_{x_i}$ of width $k-1$. From these, we construct a path decomposition of width k for S^k as follows:

- The first bags are those from the decompositions of each $S^k_{u_i}$ but with u added to each.
- The next bag consists of x and u.
- The next bags are those from the decompositions of each $S^k_{x_i}$ but with x added to each.
- The next bag consists of x and v.
- The next bag consists x and w.
- The final bags are those from the decompositions of each $S^k_{w_i}$ but with w added to each. □

Thus, by Lemmas 11 and 12, the upper bound given in Lemma 9 is optimal for each S^k.

Corollary 13 *For each* $h \geq 2$*, there exists a tree* T *with* $\mathrm{pw}(T) \leq h$ *for which every short layered drawing occupies at least* $2h-1$ *layers.*

The lower bound for short layered drawings has already been proved by Felsner *et al.* in [8], but we reproduce our own proof here to motivate a similar lower bound proof for unconstrained layered drawings later. We prove that if a graph G has a short layered drawing then that drawing occupies at least $\mathrm{pw}(G)$ layers. Our proof involves constructing a path decomposition of G with width h from a short h-layer planar drawing of G. The first bag in the decomposition contains the left-most vertices on each layer in the drawing, and we show that we can construct each successive bag by removing one vertex v from the current bag and adding a new vertex immediately to the right of v in the drawing.

To do this, we require a few definitions and preliminary results. Given a vertex v in a short h-layer drawing of G, we use $\mathrm{R}(v)$ to denote the set of vertices on the same layer but to the right of x:

$$\mathrm{R}(v) = \{u \mid u \in V(G),\ \mathrm{Y}(v) = \mathrm{Y}(u),\ \mathrm{X}(v) < \mathrm{X}(u)\}.$$

We use (v) to denote the vertex in $\mathrm{R}(v)$ closest to v; that is, (v) is the vertex in $\mathrm{R}(v)$ with the minimum x-coordinate. If $\mathrm{R}(v)$ is empty then (v) is undefined. Given a set of vertices $S \subseteq V(G)$ with exactly one vertex on each layer, we use $\mathrm{R}(S)$ to denote the set of vertices $\bigcup_{v \in S} \mathrm{R}(v)$. Finally, we use $\mathrm{F}(S)$ to denote the set of *frontier* vertices in S; that is, the vertices $v \in S$ such that $\mathrm{R}(v) \neq \emptyset$ and v has no neighbors in $\mathrm{R}(S)$ on a different layer:

$$\mathrm{F}(S) = \{v \mid v \in S,\ \mathrm{R}(v) \neq \emptyset,\ (u, v) \in E(G) \text{ and } u \in \mathrm{R}(S) \Rightarrow u \in \mathrm{R}(v)\}.$$

When we construct our path decomposition, we use $\mathrm{F}(S)$ to determine which vertex to remove from the current bag in the decomposition and which vertex to add in order to construct the next bag in the sequence.

Lemma 14 *In a short h-layer drawing of a graph G, if $S \subseteq V(G)$ has exactly one vertex on each layer and $\mathrm{R}(S) \neq \emptyset$ then $\mathrm{F}(S) \neq \emptyset$.*

Proof. Assume that $\mathrm{R}(S) \neq \emptyset$ and let $S' \subseteq S$ be the set of vertices $v \in S$ for which $\mathrm{R}(v) \neq \emptyset$; that is, $S' = \{v \mid v \in S,\ \mathrm{R}(v) \neq \emptyset\}$. In addition, let $S'' \subseteq S'$ be the set of vertices $v \in S'$ having a neighbor in $\mathrm{R}(S)$ on layer $\mathrm{Y}(v) - 1$; that is, $S'' = \{v \mid v \in S',\ \exists (v, v') \in E(G),\ v' \in \mathrm{R}(S),\ \mathrm{Y}(v') = \mathrm{Y}(v) - 1\}$. We observe that S' is not empty because $\mathrm{R}(S)$ is not empty and consider two cases:

1. $S'' = \emptyset$. In this case, the vertex v in S' with the largest y-coordinate has no neighbors in $\mathrm{R}(S)$ on layers $\mathrm{Y}(v) - 1$ or $\mathrm{Y}(v) + 1$; therefore, v belongs to $\mathrm{F}(S)$.

2. $S'' \neq \emptyset$. Let v be the vertex in S'' with the smallest y-coordinate. Thus, v has a neighbor v' in $\mathrm{R}(w)$ for some $w \in S'$ with $\mathrm{Y}(v') = \mathrm{Y}(w) = \mathrm{Y}(v) - 1$. Vertex w does not belong to S'' because v has the smallest y-coordinate of any vertex in S''; consequently, w has no neighbors in $\mathrm{R}(S)$ on layer $\mathrm{Y}(w) - 1$. Vertex w also has no neighbor in $\mathrm{R}(S)$ on layer $\mathrm{Y}(w) + 1 = \mathrm{Y}(v)$ because such an edge would cross edge (v, v'). Therefore, w belongs to $\mathrm{F}(S)$. $\square$

Finally, we obtain our lower bound:

Lemma 15 *If a graph G has a short layered drawing then that drawing occupies at least $\mathrm{pw}(G)$ layers.*

Proof. We show that, given a short h-layer drawing of G, we can construct a path decomposition of G with width h. When $|V| \leq h$, a path decomposition consisting of a single bag containing all vertices in G is sufficient. We assume, then, that $|V| > h$.

We construct the path decomposition $B_1,\ B_2,\ \ldots,\ B_{|V|-h}$ from a sequence of sets $S_0,\ S_1,\ \ldots,\ S_{|V|-h}$. More specifically, we let each $B_i = S_i \cup S_{i-1}$. We define each S_i inductively as follows:

- S_0 is the set of h vertices with minimum x-coordinates on each layer.

- We define S_i in terms of S_{i-1}. By Lemma 14, there exists at least one vertex in $\mathrm{F}(S_{i-1})$ for $1 \leq i \leq |V| - h - 1$ so we let v be the vertex with the smallest y-coordinate; thus, $S_i = (S_{i-1} \setminus \{v\}) \cup \{(v)\}$.

If we let each $B_i = S_i \cup S_{i-1}$ then we claim that $B_1,\ B_2,\ \ldots,\ B_{|V|-h}$ is a path decomposition of G with width h. The first bag B_1 contains $h+1$ vertices, and each successive bag contains exactly one vertex not found in the bags before it so $B_1 \cup B_2 \cup \ldots \cup B_{|V|-h} = V(G)$. Now consider an edge $(u, v) \in E(G)$ such that B_i is the first bag containing u and B_k is the first bag containing v, for $i \leq k$. Thus, we have $u \in S_i$, $v \in S_k$, and $v \in \mathrm{R}(S_j)$ for each S_j, $i \leq j \leq k-1$. Consequently, u belongs to each S_j and most importantly to S_{k-1}; thus, u and v both belong to B_k. Finally consider a vertex $v \in B_i \cap B_k$, for $i < k$. In other words, v is in S_i and S_{k-1} so in fact $v \in S_j$ for each $i \leq j \leq k-1$; thus, v also belongs to each bag B_j. $\square$

We cannot improve on this lower bound because the nearly complete ternary tree of depth $d \geq 1$ has pathwidth d by Lemma 4 and a short d-layer drawing. To obtain such a drawing, we simply place vertices at depth i on layer i except for each leaf which we place next to its parent and on the same layer.

Lemma 16 *For each $h \geq 1$, there exists a graph G with $\mathrm{pw}(G) \leq h$ and a short h-layer drawing.*

By Lemma 9 and Corollary 13, and Lemmas 15 and 16, then, our bounds on the number of layers in short layered drawings of trees are optimal:

Theorem 1 *For each $h \geq 2$, the lower bound h and the upper bound $2h-1$ are optimal bounds on the number of layers in short layered drawings of trees with pathwidth h.*

4 Proper Layered Drawings

As with short layered drawings, we obtain our upper bound by constructing drawings. This bound was first proved in [5]:

Lemma 17 *Every tree T with $\mathrm{pw}(T) \geq 2$ has a proper $(3\mathrm{pw}(T)-2)$-layer drawing.*

Proof. We obtain a proper $(3\mathrm{pw}(T)-2)$-layer drawing of any tree T with $\mathrm{pw}(T) \geq 2$ by first drawing a main path P of T on the top two layers. Each component T' in $T \setminus P$ has pathwidth at most $\mathrm{pw}(T)-1$. Let v be the vertex in T' adjacent to a vertex w in P. We insert a drawing of each T' onto the $(3\mathrm{pw}(T)-4)$ layers below $w \in P$ and then draw the missing edge (v, w) as a straight line between v and w. To avoid crossings, we draw T' so that v lies on the layer immediately below w; that is, v is exposed in the drawing of T'. It remains to prove, then, that such drawings of T' exist; that is, we must prove that for any tree T with $\mathrm{pw}(T) \geq 1$ and vertex $v \in T$, there exists a proper $(3\mathrm{pw}(T)-1)$-layer drawing of T in which v is exposed.

Let $P = v_1 v_2 \ldots v_n$ be a main path of T. The proof is by induction on $\mathrm{pw}(T)$. For $\mathrm{pw}(T) = 1$, there exists a proper 2-layer drawing of T by Lemma 8. Clearly v is exposed in this drawing since there are only two layers. Now suppose that $\mathrm{pw}(T) \geq 2$, and let R be the path between v and a vertex v_i on P. We begin drawing T on $(3\mathrm{pw}(T)-1)$ layers by first drawing the path $R v_i v_{i-1} \ldots v_1$ on layers one and two and then the path $v_{i+1} v_{i+2} \ldots v_n$ on layers two and three below edge (v_i, v_{i-1}). Each connected component T' of $(T \setminus P) \setminus R$ has pathwidth at most $\mathrm{pw}(T) - 1$. Let v' be the vertex in T' adjacent to a vertex $w \in P \cup R$. By induction, there exists a proper $(3\mathrm{pw}(T) - 4)$-layer drawing of T' in which v' is exposed. We insert this drawing onto the layers below w. The final drawing is illustrated in Figure 3. □

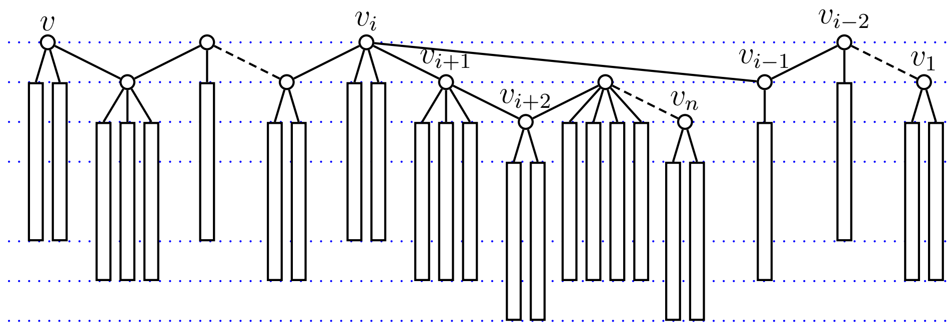

Figure 3: A proper $(3\mathrm{pw}(T) - 1)$-layer drawing T in which vertex v in T is exposed.

Once again, we prove that this upper bound is optimal for set of recursively-defined rooted trees. We define P^1 to be the tree consisting of a single edge and, for $k \geq 2$, we define P^k to be just like S^k except that:

- we attach another child y to the root v that has two children and each of those children has exactly one child leaf;
- we make use of Lemma 10 when $i = 2$ by giving u, w and x each $n = 21$ children. Each child is the root of a subtree isomorphic to P^{k-1}.

See Figure 4.

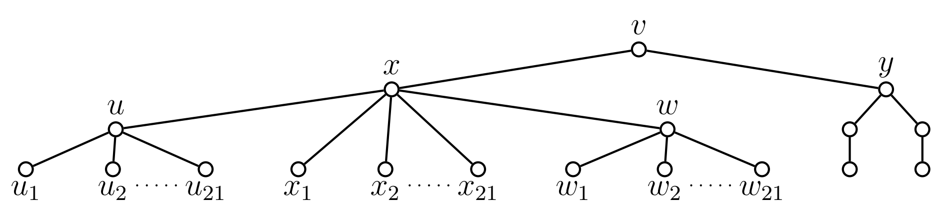

Figure 4: Tree P^k for $k \geq 2$. Each u_i, w_i and x_i is the root of a subtree isomorphic to P^{k-1}.

Lemma 18 *For $k \geq 1$, every proper drawing of P^k occupies at least $3k - 2$ layers and at least $3k - 1$ layers if v is exposed.*

Proof. The proof is by induction on k. For $k = 1$, tree $P^k = P^1$ consists of a single edge so every proper layered drawing of P^k occupies at least $3k - 1 = 2$ layers.

Assume that $k \geq 2$. By induction, every proper layered drawing of P^{k-1} occupies at least $3(k-1)-2 = 3k-5$ layers and $3k-4$ layers if the root is exposed. By Lemma 10, then, any proper layered drawing of subtree P_u^k, P_w^k or $(P_x^k \setminus P_u^k) \setminus P_w^k$ occupies at least $3k-3$ layers and if exactly $3k-3$ layers then the root (u, w or x) lies on the top or bottom layer. In addition, if the drawing uses exactly $3k-2$ layers then the root lies on one of the two topmost or bottommost layers. If we have a proper $(3k-3)$-layer drawing of P^k then vertex u, w or x is not on the top or bottom layer and therefore is not exposed in the drawing of its corresponding subtree. However, in that case, the corresponding subtree occupies at least $3k-2$ layers so the drawing of P^k occupies at least $3k-2$ layers.

If v is exposed on the top layer in a proper layered drawing of P^k then either u or w is on layer three. Assume without loss of generality that w is on layer three. See Figure 5. For $k=2$, $P_w^k = P_w^2$ contains no vertices on the top layer. This is because subtrees P_u^k and P_y^k have vertices on layers 1, 2 and 3; thus, P_w^k would need a path of length 4 starting at w to reach the top layer. The longest such path has length 2. If P_w^k occupies exactly $3k-3=3$ layers then it occupies layers 3, 4 and 5; otherwise, it occupies at least $3k-2=4$ layers, layers 2 to 5. Therefore, the drawing of P^k occupies at least $3k-1=5$ layers. For $k \geq 3$, every drawing of P_w^k occupies at least $3k-3 \geq 6$ so if it occupies exactly $3k-3$ layers then it occupies layers 3 to $3k-1$. If it occupies exactly $3k-2$ layers then it occupies layers 2 to $3k-1$ and if more than $3k-2$ then layers 1 to $3k-1$. $\square$

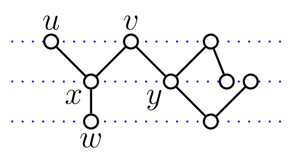

Figure 5: Vertex v is exposed on the top layer in a proper layered drawing of P^k.

The pathwidth of each P^k is identical to that of S^k.

Lemma 19 *For $k \geq 1$, $\mathrm{pw}(P^k) \leq k$.*

Thus, by Lemmas 18 and 19, the upper bound given in Lemma 17 is optimal for each P^k.

Corollary 20 *For each $h \geq 2$, there exists a tree T with $\mathrm{pw}(T) \leq h$ for which every proper layered drawing occupies at least $3h-2$ layers.*

For proper layered drawings, our lower bound is one layer larger than for short layered drawings because we do not permit flat edges. Our proof proceeds as in Lemma 15 except that we let each bag $B_i = S_i$. This is possible because we do not permit flat edges. Thus, we have the following lower bound for proper drawings of graphs.

Lemma 21 *If a graph G has a proper layered drawing then that drawing occupies at least $\mathrm{pw}(G)+1$ layers.*

We cannot improve this lower bound because the complete ternary tree of depth d has pathwidth d by Lemma 4 and a proper $d+1$-layer drawing. We simply place each vertex at depth i on layer $i+1$.

Lemma 22 *For each $h \geq 0$, there exists a graph G with $\mathrm{pw}(G) \leq h$ and a proper $(h+1)$-layer drawing.*

By Lemma 17 and Corollary 20, and Lemmas 21 and 22, then, our bounds on the number of layers in proper layered drawings of trees are optimal:

Theorem 2 *For each $h \geq 2$, the lower bound h and the upper bound $3h-2$ are optimal bounds on the number of layers used in proper layered drawings of trees with pathwidth h.*

5 Upright and Unconstrained Layered Drawings

In this section we first state and prove an upper bound for *upright* layered drawings and then prove that this bound is the optimal upper bound for *unconstrained* layered drawings. Because every upright layered drawing is by definition an unconstrained layered drawing, we will have thus proven that our upper bound is optimal for both upright and unconstrained layered drawings.

First we prove the upper bound:

Lemma 23 *Every tree T with $\mathrm{pw}(T) \geq 1$ has an upright $\lceil 3\mathrm{pw}(T)/2 \rceil$-layer drawing.*

Proof. We actually prove that if v is a vertex in a tree T then there is an upright $\lceil 3\mathrm{pw}(T)/2 \rceil$-layer drawing of T in which v is exposed. The proof is by induction on the pathwidth of T. If $\mathrm{pw}(T) = 1$ then, by Lemma 8, T has an upright 2-layer drawing.

Assume that $\mathrm{pw}(T) \geq 2$. We begin by drawing $P = v_1 v_2 \dots v_m$, a main path of T, on $\lceil 3\mathrm{pw}(T)/2 \rceil$ layers so that its vertices alternate between the top and bottom layers. If v lies on P then v will be exposed in the final drawing. Otherwise, v belongs to a component T' of $T \setminus P$. Let P' be a main path of T', v' the vertex in T' adjacent to a vertex v_i in P and let $Q = w_1 w_2 \dots w_n$ be a path from v' to an end-vertex of P' such that Q intersects the path from v to v'. For example, see Figure 6.

We continue by drawing Q on $\lceil 3\mathrm{pw}(T)/2 \rceil - 1$ layers next to v_i so that its vertices alternate between the top and bottom of these layers. If v belongs to Q then we draw Q so that v is exposed in the final drawing. Otherwise, v belongs to a component T'' of $T' \setminus P'$. Let v'' be the vertex in T'' adjacent to a vertex w_j of Q and R the path from v'' to v. In this case we draw Q so that w_j is not exposed so that we can expose v. We then draw path R on $\lceil 3\mathrm{pw}(T)/2 \rceil - 2$ layers next to w_j so that its vertices alternate between the top and bottom of these layers and v is exposed. If Q does not contain all of P' and vertex w_k in Q is adjacent to a vertex in $P' \setminus Q$ then we draw $P' \setminus Q$ on the $\lceil 3\mathrm{pw}(T)/2 \rceil - 2$ layers next to w_k so that its vertices alternate between the top and bottom of these layers. Figure 6 illustrates the relationships between paths P, P', Q and R in T.

Each component C of $(T \setminus T') \setminus P$ has pathwidth at most $\mathrm{pw}(T) - 1$ so, by induction, each has a $(\lceil 3\mathrm{pw}(T)/2 \rceil - 1)$-layer drawing in which the vertex in C adjacent to a vertex

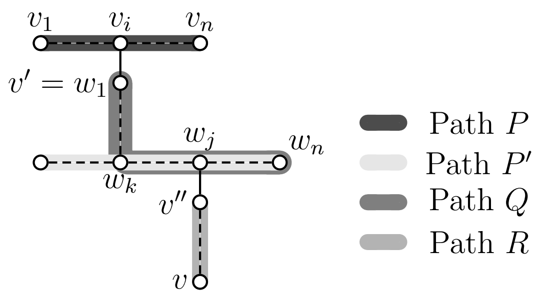

Figure 6: Paths P, P', Q and R in a tree T.

in P is exposed. We recursively obtain such a drawing and insert it into the drawing next to the appropriate vertex in P. Similarly, each component of $T' \setminus P'$ and therefore each component C of $T' \setminus P' \setminus Q \setminus R$ has pathwidth at most $\mathrm{pw}(T) - 2$ so by induction each has a $(\lceil 3\mathrm{pw}(T)/2 \rceil - 3)$-layer drawing in which the vertex in C adjacent to a vertex in $P' \cup Q \cup R$ is exposed. We recursively obtain such a drawing and insert into the drawing next to the appropriate vertex in $P' \cup Q \cup R$. The final result is illustrated in Figure 7. The rectangles represent drawings of components of $T \setminus P \setminus P' \setminus Q \setminus R$. $\square$

We now prove that the upper bound just proven is optimal for unconstrained layered drawings. As for short and proper layered drawings, we prove optimality by describing a set of rooted trees whose unconstrained layered drawings use the number of layers given in the upper bound. These trees have the property that, when drawn on a minimum number of layers, the root is not accessible. A vertex v is *accessible* in a layered drawing Γ if we can insert a layered drawing of some path P into Γ without creating crossings so that one end vertex is adjacent to v and the other is exposed. Notice that if a vertex is exposed then it is accessible but the reverse is not always true. The next two lemmas show how we prevent the root from being accessible in a minimum layer drawing.

Lemma 24 *Let T be a tree rooted at v, and let u and w be children of v. Let Γ be an unconstrained h-layer drawing of T for $h \geq 1$ in which subtrees T_u and T_w each occupy at least $h-1$ layers. Then, Γ contains a drawing of T_u in which u is accessible.*

Proof. There is a path in the drawing of T outside T_u that begins at v and ends at an exposed vertex in T_w; therefore, u is accessible in the drawing of T_u. $\square$

Lemma 25 *Let u and w be vertices in a tree T rooted at vertex v such that v is the only common ancestor of u and w. Assume that u_1 and u_2 are children of u such that any unconstrained layered drawing of subtree T_{u_i} occupies at least $h \geq 0$ layers but u_i is not accessible in any h-layer drawing. Similarly, assume that w_1 and w_2 are children of w such that any drawing of subtree T_{w_i} occupies at least $h+1$ layers but w_i is not accessible in any $(h+1)$-layer drawing. Then there are no unconstrained $(h+2)$-layer drawings of T in which v is accessible.*

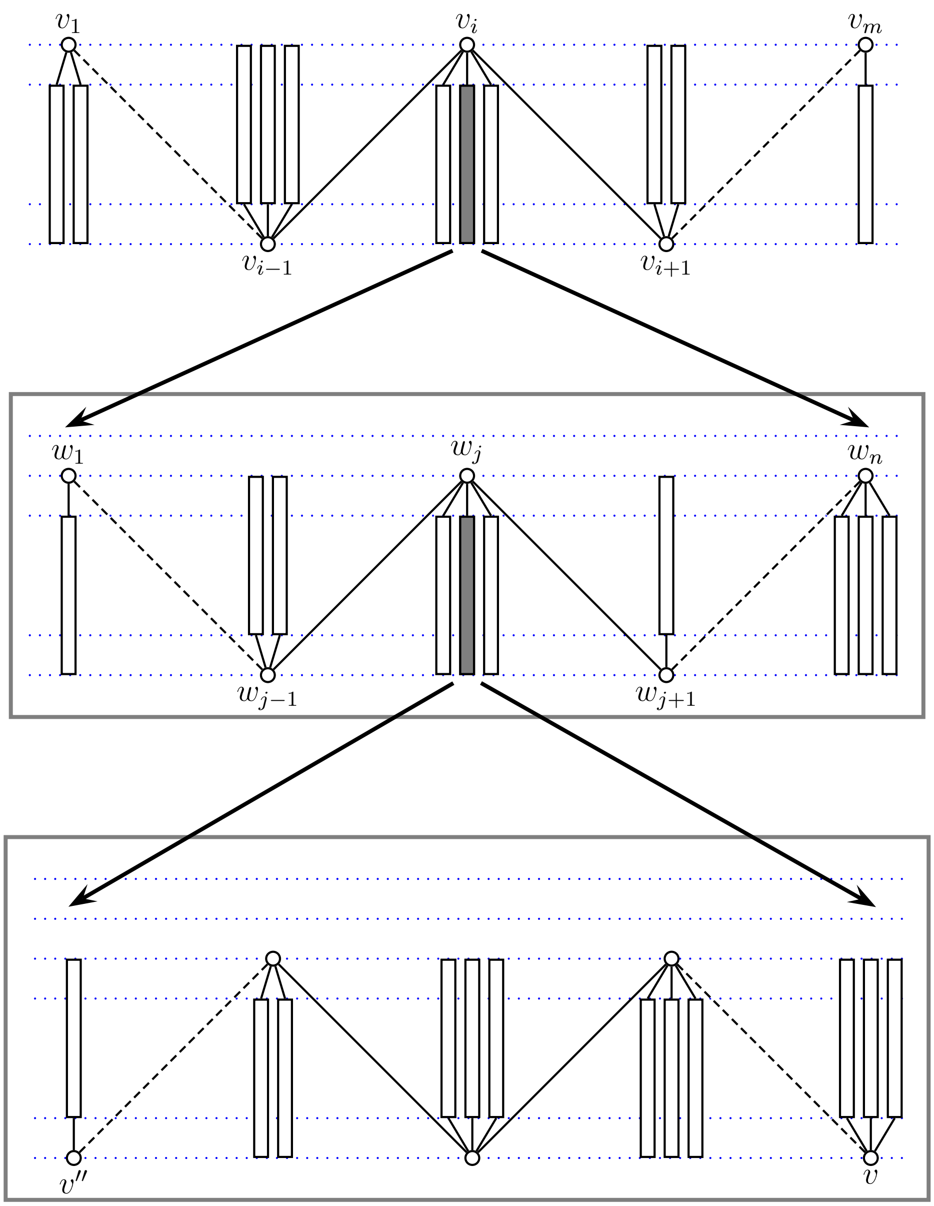

Figure 7: An upright ($\lceil 3\mathrm{pw}(T)/2 \rceil$)-layer drawing of T (top), an upright ($\lceil 3\mathrm{pw}(T)/2 \rceil - 1$)-layer drawing of T' (middle) and an upright ($\lceil 3\mathrm{pw}(T)/2 \rceil - 2$)-layer drawing of T'' (bottom).

Proof. Assume by way of contradiction that we have an unconstrained ($h+2$)-layer drawing Γ of T in which v is accessible. We thus insert the drawing of a path $P = v_1 v_2 \ldots v_n$ into Γ so that v_1 is adjacent to v and v_n is on the top layer. We let $T' = T \cup P$.

We claim that, in this drawing, $T' \setminus T'_w$ occupies the top $h+1$ layers and $(T' \setminus T'_w) \setminus T'_u$ occupies the top h layers. Each subtree T'_{w_i} occupies at least $h+1$ layers so, by Lemma 24, w_i is accessible in the drawing of T'_{w_i}. Therefore, each T'_{w_i} occupies at least $h+2$ layers. By Lemma 1, then, $T' \setminus T'_w$ occupies at most $h+1$ layers. Similarly, each subtree T'_{u_i} occupies at least h of these layers so, by Lemma 24, u_i is accessible in the drawing of T'_{u_i}. Therefore, each T'_{u_i} occupies at least $h+1$ layers so, by Lemma 1, $(T' \setminus T'_w) \setminus T'_u$ occupies at most h layers. Since $v_n \in P$ lies on the top layer and $v_n \in T' \setminus T'_w$, subtree $T' \setminus T'_w$ occupies the top $h+1$ layers, that is layers $1, 2, \ldots, h+1$. Similarly, we have $v_n \in (T' \setminus T'_w) \setminus T'_u$ so $(T' \setminus T'_w) \setminus T'_u$ occupies the top h layers.

Let T'' be a subdivision of T' created by subdividing each long edge that crosses layer $h+1$ in Γ. We obtain an $(h+2)$-layer drawing Γ' of T'' from Γ by placing the new vertices in T'' on layer $h+1$ where the edges they subdivide intersect layer $h+1$. Let S be the non-empty set of vertices in T'' on layer $h+2$ and T''' the connected component in $T'' \setminus S$ containing u. Thus, Γ' contains an $(h+1)$-layer drawing Γ'' of T'''. However, this contradicts Lemma 1 because $T''' \setminus u$ contains three components that occupy $h+1$ layers in Γ''. The component containing u_1 occupies $h+1$ layers because we showed above that T'_{u_1} occupies the top $h+1$ layers. The same applies to the component containing u_2. The component containing v also contains $v_n \in P$ on the top layer because we showed above that P lies on the top h layers. This component also contains a vertex on layer $h+1$ adjacent to a vertex in S; therefore, the component containing v occupies all $h+1$ layers. □

Using Lemma 25, we recursively define a tree T^k for each $k \geq 0$. Tree T^0 is the empty tree, and tree T^1 is the single vertex. For $k \geq 2$, tree T^k consists of a root v with two children u and w. Child u has two children u_1 and u_2, each roots of subtrees isomorphic to T^{k-2}. Similarly, child w has two children w_1 and w_2, each roots of subtrees isomorphic to T^{k-1}. These trees are illustrated in Figure 8. As expected, we obtain the following result:

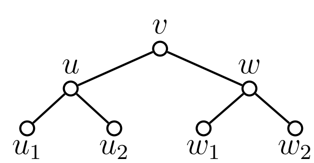

Figure 8: Tree T^k, for $k \geq 2$. Each u_i is the root of a subtree isomorphic to T^{k-2} and each w_i the root of a subtree isomorphic to T^{k-1}.

Lemma 26 *Every unconstrained layered drawing of T^k occupies at least $k-1$ layers and at least k layers if v is accessible.*

PROOF. Every layered drawing of tree T^1 occupies at least 1 layer. Tree T^2 contains a complete ternary tree of height 1 so by Corollary 2 every drawing of T^2 occupies at least 2 layers. For $k \geq 3$, then, the result follows by induction and Lemmas 24 and 25. □

Next we obtain an upper bound on the pathwidth of each tree T^k.

Lemma 27 *For $k \geq 0$,* $\text{pw}(T^k) \leq \lfloor 2k/3 \rfloor$

PROOF. The proof is by induction on k. We first prove that for $k' \geq 0$ and $k \geq 2$, if T^{k-2} contains no k'-critical vertices and we have $\text{pw}(T^{k-1}) \leq k'$ then T^k contains no $(k'+1)$-critical vertices, T^{k+1} contains no $(k'+2)$-critical vertices and $\text{pw}(T^{k+2}) \leq k'+2$.

Subtrees $T^k_{w_1}$ and $T^k_{w_2}$ are isomorphic to T^{k-1} so they each have pathwidth at most k'. Therefore, subtree T^k_w contains no $(k'+1)$-critical vertices and neither does subtree T^k_u since it is isomorphic to a subtree of T^k_w. Subtrees $T^{k+1}_{u_1}$ and $T^{k+1}_{u_2}$ are isomorphic to T^{k-2} so, by Lemma 5, subtree T^k_u has pathwidth at most k' and therefore v is not $(k'+1)$-critical. Thus, T^k contains no $(k'+1)$-critical vertices.

Subtree T^{k+1}_u is isomorphic to T^k_w so it contains no $(k'+1)$-critical vertices. By Lemma 5, T^{k+1}_u has pathwidth at most $k'+1$ so v is not $(k'+2)$-critical. Subtrees $T^{k+1}_{w_1}$ and $T^{k+1}_{w_2}$ are isomorphic to T^k so they contain no $(k'+1)$-critical vertices. By Lemma 5, then, subtree T^{k+1}_w has pathwidth at most $k'+1$ and therefore contains no $(k'+2)$-critical vertices. Thus, T^{k+1} contains no $(k'+2)$-critical vertices.

Subtree T^{k+2}_u is isomorphic to T^{k+1}_w so it has pathwidth at most $k'+1$. Subtrees $T^{k+2}_{w_1}$ and $T^{k+2}_{w_2}$ contain no $(k'+2)$-critical vertices because they are isomorphic to T^{k+1}. By Lemma 5, then, T^{k+2} has pathwidth at most $k'+2$.

The lemma then follows by induction on k because for $k=0$ trees $T^k = T^0$ and $T^{k+1} = T^1$ both have pathwidth 0. □

Finally, we show that the upper bound of Lemma 23 is optimal:

Lemma 28 *For each $h \geq 0$, there exists a tree T with $\text{pw}(T) \leq h$ for which every unconstrained layered drawing occupies at least $\lceil 3h/2 \rceil$ layers.*

PROOF. Consider the tree T rooted at r having three children each the roots of subtrees isomorphic to $T^{\lceil 3h/2 \rceil - 1}$. By Lemma 27, tree $T^{\lceil 3h/2 \rceil - 1}$ has pathwidth at most $h-1$. Therefore, by Lemma 5, T has pathwidth at most h.

By Lemma 26, in any drawing of T, the three subtrees occupy at least $\lceil 3h/2 \rceil - 2$ layers. By Lemma 1, then, T occupies at least $\lceil 3h/2 \rceil - 1$ layers. However, if T occupies exactly $\lceil 3h/2 \rceil - 1$ layers then each subtree has a vertex on either the top or bottom layer implying that the root of each subtree is accessible. Therefore, by Lemma 26, each subtree occupies $\lceil 3h/2 \rceil - 1$ layers. Consequently, by Lemma 1, every layered drawing of T occupies at least $\lceil 3h/2 \rceil$ layers. □

The optimal lower bounds for upright and unconstrained layered drawings differ. However, both use the following result similar to Lemma 14 for short layered drawings:

Lemma 29 *In an unconstrained h-layer drawing of a graph G, if $S \subseteq V(G)$ has exactly one vertex on each layer and $\text{R}(S) \neq \emptyset$ then $\text{F}(S) \neq \emptyset$.*

PROOF. Suppose that $\text{R}(S) \neq \emptyset$ and let $S' \subseteq S$ be the set $\{v \mid v \in S,\ \text{R}(v) \neq \emptyset\}$. We prove that $\text{F}(S) \neq \emptyset$ by showing that at least one vertex $v \in S'$ belongs to $\text{F}(S)$. By definition, if $v \in \text{F}(S)$ then v has no neighbor in $\text{R}(S) = \text{R}(S')$ on a different layer. To find v, we construct a sequence of vertices $v_1, v_2, \ldots, v_p$ on a subset of S' in which $v = v_p$ is the last vertex. The sequence satisfies the following constraints:

1. For each pair of edges $e = (v_j, u)$ and $e' = (v_i, w)$ where $i < j$ and $u, w \in \mathrm{R}(S)$, if e and e' cross the same layer l, then e crosses l to the left of e'; and,

2. v_p has no neighbors in $\mathrm{R}(S') \setminus \mathrm{R}(\{v_1, v_2, \ldots, v_p\})$

If such a sequence exists then the last vertex v_p does not have any neighbors in $\mathrm{R}(S')$ on a different layer; therefore, v_p belongs to $\mathrm{F}(S)$.

We construct such a sequence inductively. We let v_1 be any vertex in S' and, then, given v_i, we let v_{i+1} be a vertex in $S' \setminus \{v_1, v_2, \ldots, v_i\}$ such that v_i has a neighbor in $\mathrm{R}(v_{i+1})$. If no such v_{i+1} exists then we can take $v_i = v_p = v$.

We prove that the constructed sequence satisfies the first constraint given above by way of contradiction. Suppose that v_j is the first vertex to violate the first constraint; that is, suppose that there is some edge $e = (v_j, u)$ with $u \in \mathrm{R}(S)$ that crosses a layer l to the right of some edge $e' = (v_i, w)$, where $i < j$ and $w \in \mathrm{R}(S)$. We split the drawing into two regions with the horizontal ray anchored at v_j pointing in the negative x-direction, the segment from v_j to the point p where e intersects layer l, and the horizontal ray anchored at p pointing in the positive direction. These two regions are illustrated in Figure 9. The boundary between them is highlighted by the thick gray polygonal line. By definition, v_{j-1} has a neighbor

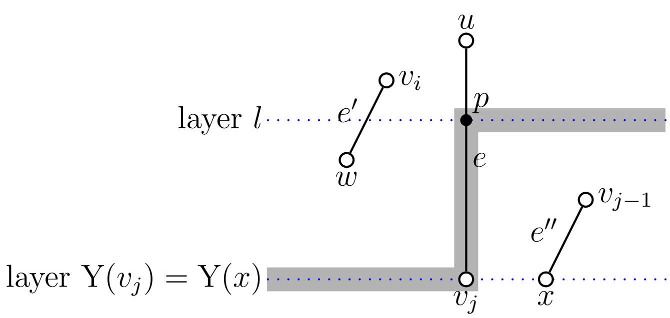

Figure 9: Edges e and e' define two regions.

$x \in \mathrm{R}(v_j)$. We call that edge e'' and claim that e' and e'' lie in opposite regions. Firstly, e' does not cross layer $\mathrm{Y}(v_j)$, for otherwise, it crosses to the left of v_j while e'' crosses to the right. In this case, v_{j-1} violates the first constraint given above contradicting the minimality of j. Secondly, edge e'' does not cross layer l for otherwise it crosses to the right of p while e' crosses to the left. Once again, v_{j-1} violates the first constraint given above. Thus, e' and e'' lie entirely in opposite regions. Now let v_k be the first vertex after v_i that does not lie in the same region as v_i. Thus, v_{k-1} lies in the same region as v_i and has a neighbor y in $\mathrm{R}(v_k)$. Consequently, edge $e''' = (v_{k-1}, y)$ crosses the boundary between the two regions. However, it cannot cross layer $\mathrm{Y}(v_j)$ left of v_j because e'' crosses to the right. In this case, v_{j-1} violates the first constraint, contradicting the minimality of j. Similarly, it cannot cross layer l to the right of p because e' crosses to the left. In this case, v_{k-1} violates the first constraint, contradicting the minimality of j. Thus, we have shown that the constructed sequence satisfies the first constraint.

The second constraint is also satsified because S' is finite. Consequently, vertex v_p belongs to $\mathrm{F}(S)$. □

For unconstrained layered drawings, the optimal lower bound is the same as for short layered drawings. The proof is similar to the proof of Lemma 15 for short layered drawings. We simply change all references to short layered drawings to unconstrained layered drawings and all references to Lemma 14 to Lemma 29. This bound is also proved independently by Felsner *et al.* in [9].

Lemma 30 *For every graph G, any unconstrained layered drawing of G occupies at least $\mathrm{pw}(G)$ layers.*

Short layered drawings are unconstrained layered drawings so, by Lemma 16, our lower bound is optimal.

Corollary 31 *For each $h \geq 1$, there exists a graph G with $\mathrm{pw}(G) \leq h$ and an unconstrained h-layer drawing.*

Thus, by Lemmas 23, 28 and 30, and Corollary 31, then, our bounds on the number of layers in unconstrained layered drawings of trees are optimal:

Theorem 3 *For each $h \geq 1$, the lower bound h and the upper bound $\lceil 3h/2 \rceil$ are optimal bounds on the number of layers in unconstrained layered drawings of trees with pathwidth h.*

For upright layered drawings, we observe that in the proof of Lemma 30 we can simply let each $B_i = S_i$ because flat edges are not permitted in the drawing.

Lemma 32 *If a graph G has an upright layered drawing then that drawing occupies at least $\mathrm{pw}(G) + 1$ layers.*

The optimality of this bound follows from Lemma 22 because every proper layered drawing is by definition an upright layered drawing.

Corollary 33 *For each $h \geq 0$, there exists a graph G with $\mathrm{pw}(G) \leq h$ and an upright $(h + 1)$-layer drawing.*

Thus, by Lemmas 23, 28 and 32, and Corollary 33, then, our bounds on the number of layers in upright layered drawings of trees are optimal:

Theorem 4 *For each $h \geq 1$, the lower bound $h+1$ and the upper bound $\lceil 3h/2 \rceil$ are optimal bounds on the number of layers in upright layered drawings of trees with pathwidth h.*

6 Linear-Time Drawing Algorithms

We can obtain the layered drawings described in the proofs of Lemmas 9, 17 and 23 in linear time. Each drawing depends on there being an algorithm that can efficiently decompose a tree into one or more paths and subtrees. More specifically, given a vertex v in a tree T, the drawings depend on three different decompositions:

1. a main path P of T and the components of $T \setminus P$;
2. a main path P of T, the path R from v to P, and the components of $(T \setminus P) \setminus R$; and,
3. a main path P of T, a main path P' of the subtree in $T \setminus P$ containing v, the path Q from P to an end vertex of P' such that Q intersects the path from v to P, the path R from v to Q, and the components of $(((T \setminus P) \setminus P') \setminus Q) \setminus R$.

Recall that the first decomposition is applied initially to the whole tree and then the last two decompositions are recursively applied to the subtrees until the entire tree is decomposed into paths. We describe an algorithm that can accomplish this in linear time.

As a preprocessing step, we root the tree at an arbitrary vertex and then apply the linear-time algorithm of [7] for finding the pathwidth of a tree. More specifically, given a tree T the algorithm computes a label for each a vertex v in the tree. The label consists of a sequence of non-negative integers $(a_1,\ a_2,\ \ldots a_p)$ in descending order and a corresponding sequence of vertices $(v_1,\ v_2,\ \ldots v_p)$ in the subtree T_v rooted at v such that:

1. $\mathrm{pw}(T_v) = a_1$;
2. for $1 \leq i \leq p-1$, $\mathrm{pw}(T_v \setminus T_{v_1} \setminus T_{v_2} \setminus \ldots \setminus T_{v_{i-1}}) = a_{i+1}$; and
3. for $1 \leq i \leq p-1$, v_i is an a_i-critical vertex in $T_v \setminus T_{v_1} \setminus T_{v_2} \setminus \ldots \setminus T_{v_{i-1}}$.

For our algorithm we do not need to save the whole label for each vertex v. Instead, we need only retain the values a_1 and a_2 and corresponding vertices v_1 and v_2. We refer to v_1 with $\mathrm{cr}_1(v)$, v_2 with $\mathrm{cr}_2(v)$, a_1 with $\mathrm{pw}_1(v)$ and a_2 with $\mathrm{pw}_2(v)$. In the case that a_2 and v_2 do not exist, we simply say that $\mathrm{cr}_2(v)$ and $\mathrm{pw}_2(v)$ are undefined.

The first step of each decomposition is to find a main path in the tree. The next two results show how we find a main path.

Lemma 34 *Let v be a vertex in a rooted tree T. Then, there exist at most two vertices u and w that are descendants of v with $\mathrm{pw}_1(u) = \mathrm{pw}_1(w) = \mathrm{pw}_1(v)$, and each child c of u or w has $\mathrm{pw}_1(c) < \mathrm{pw}_1(v)$. Furthermore, the path between u and w contains $\mathrm{cr}_1(v)$ and is a main path for T_v.*

Proof. Suppose that there are two such descendants, u and w. Since $\mathrm{pw}_1(u) = \mathrm{pw}_1(w) = \mathrm{pw}_1(v)$, vertex u is not a descendant of w, and w is not a descendant of u. In addition, vertex $\mathrm{cr}_1(v)$ is the lowest common ancestor of u and w because, by Lemma 6, $\mathrm{cr}_1(v)$ is the unique $\mathrm{pw}_1(v)$-critical vertex in T_v.

If there is a third such descendent x of v then $\mathrm{cr}_1(v)$ is the lowest common ancestor of u, w and x, so $T_v \setminus \mathrm{cr}_1(v)$ contains three components with $\mathrm{pw}_1(v)$. By Lemma 4, however, this means that T_v has pathwidth greater than $\mathrm{pw}_1(v)$, a contradiction. Thus, there are at most two such descendents u and w.

Let P be the path from u to w, and consider a component T' of $T \setminus P$. Let x be the vertex on P that is adjacent to a vertex in T'. If $\mathrm{pw}(T') \geq \mathrm{pw}_1(v)$ then x is not a descendant of u or w. Consequently, there are at least three components in $T_v \setminus x$ with pathwidth at least $\mathrm{pw}_1(v)$: one containing T_u, another containing T_w, and the third containing T'. By

Lemma 4, however, this means that T_v has pathwidth greater than $\mathrm{pw}_1(v)$, a contradiction. Thus, P is a main path. □

Corollary 35 *To find a main path P in a tree rooted at v, we initialize P to be the single-vertex path consisting of $\mathrm{cr}_1(v)$. We then walk down the tree following edges to vertices u with $\mathrm{pw}_1(u) = \mathrm{pw}_1(v)$. As we walk, we add each such u to the appropriate end of P.*

It is convenient if the main path removed in the first decomposition contains the root. This is because each remaining component is a rooted subtree of the original tree; consequently, we can reuse the labels calculated in the preprocessing step to recursively decompose each remaining component. If the main path does not contain r then we reroot the tree at $\mathrm{cr}_1(r)$ and relabel the tree. Then, the new root r is equal to $\mathrm{cr}_1(r)$ so the main path found using the algorithm described in Corollary 35 contains $r = \mathrm{cr}_1(r)$. This decomposition is illustrated in Figure 10(a).

We then apply either decomposition two or three to the remaining subtrees. The root of each subtree is the vertex v in the subtree that is adjacent to the main path just removed.

We apply decomposition two to subtree T_v by again using the algorithm described in Corollary 35 to find a main path P of T_v. The path R from the subtree root v to P is simply the path from v to $\mathrm{cr}_1(v) \in P$. We traverse this path by walking up the tree from $\mathrm{cr}_1(v)$ to v. This decomposition is illustrated in Figure 10(b).

We apply decomposition three to subtree T_v by again using the algorithm described in Corollary 35 to find a main path P of T_v. If $v = \mathrm{cr}_1(v)$, then the decomposition is complete since paths P', Q and R have zero length. On the other hand, if $v \neq \mathrm{cr}_1(v)$ then v belongs to subtree $T' = T_v \setminus T_{\mathrm{cr}_1(v)}$. We find a main path P' of T' by again applying the algorithm described in Corollary 35 except that, for each ancestor w of $\mathrm{cr}_1(v)$, we refer to $\mathrm{cr}_2(w)$ and $\mathrm{pw}_2(w)$ instead of to $\mathrm{cr}_1(w)$ and $\mathrm{pw}_1(w)$, respectively. Let X be the path from $\mathrm{cr}_2(v)$ to v. Path Q is then composed of the path from $\mathrm{cr}_1(v)$ to the nearest vertex w in X and the path from an end vertex of P' to w. We traverse Q by walking up the tree from $\mathrm{cr}_1(v)$ to w and then from an end vertex of P' again up to w. Path R is the path from w to v. We traverse R by walking up the tree from w to v. Two cases for this decomposition are illustrated in Figure 10(c-d).

After applying decompositions two or three to T_v, we recursively apply either of these decompositions to any remaining components of T_v. Because the set of paths removed from T_v are connected and contain the root v, each of the remaining components is actually a rooted subtree of the original tree. In addition, the component vertex adjacent to the removed paths is precisely the root of the component. Consequently, we can reuse the vertex labels calculated earlier to recursively apply either of these decompositions exactly as we just described them for T_v.

We claim that the recursion finishes in linear time. The preprocessing step applies a linear-time labelling algorithm to the tree. The first decomposition may require rerooting and relabelling the tree, but both of these are accomplished in linear time. The remainder of the algorithm involves traversing various paths and collecting adjacent subtrees for further decomposition. These traversals involve visiting each vertex in the tree a small constant number of times. Thus, we have linear-time algorithms to construct the drawings described in the proofs of Lemmas 9, 17 and 23:

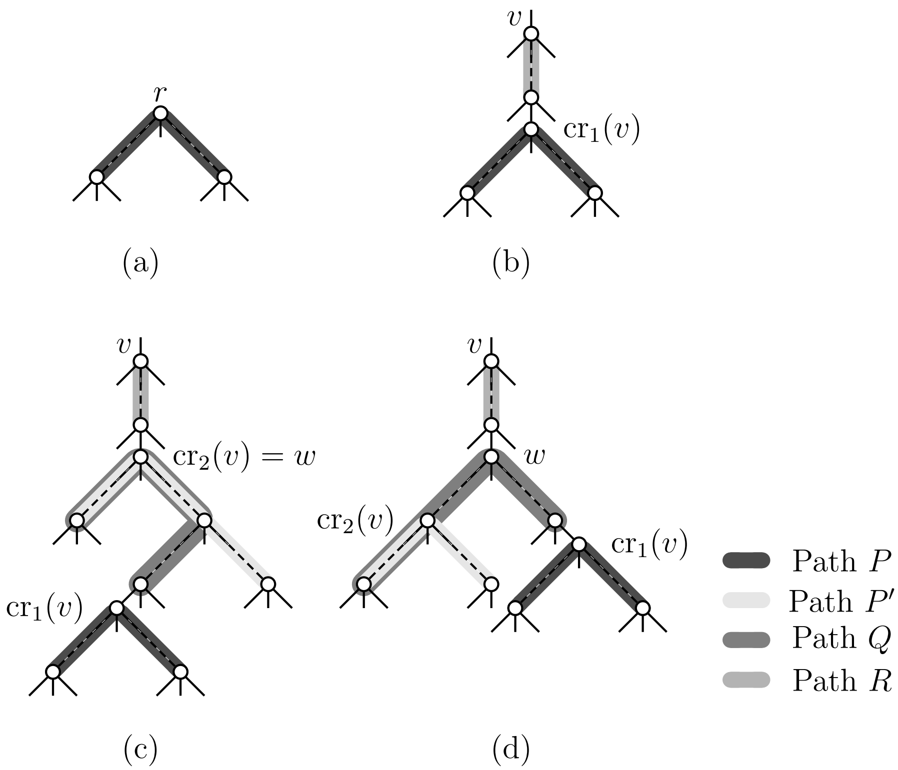

Figure 10: Decomposing a tree.

Theorem 5 *The following drawings of a tree T with $\mathrm{pw}(T) = h$ can be obtained in linear time:*

- *an unconstrained* $(\lceil 3h/2 \rceil)$*-layer drawing (if* $h \geq 1$*);*
- *an upright* $(\lceil 3h/2 \rceil)$*-layer drawing (if* $h \geq 1$*);*
- *a short* $(2h-1)$*-layer drawing (if* $h \geq 2$*); and*
- *a proper* $(3h-3)$*-layer drawing (if* $h \geq 2$*).*

7 Conclusions

In this paper we have proven optimal upper and lower bounds on the number of layers required by layered drawings of trees and given linear-time algorithms for obtaining drawings that match the optimal upper bounds.

These linear-time algorithms are significant because, even though there is an algorithm for determining whether or not an arbitrary graph has an h-layer drawing that runs in $f(h) \cdot n$ time [4], the value of $f(h)$ is too large for the algorithm to be practical for even small values of h. The results in this paper demonstrate one approach to solving this general problem, by considering increasingly more general classes of graphs. Another approach is to characterize the graphs whose drawings occupy a specific number of layers and then design efficient algorithms for determining whether or not a given graph satisfies the corresponding characterization. So far this approach has been used for only 2-layer [11, 6, 3] and 3-layer [10, 3] drawings.

8 Acknowledgements

I thank my supervisor Sue Whitesides for proof-reading and helping to improve the readability of this document. This work began as an attempt to reconstruct a proof. I thank David R. Wood for suggesting that reconstruction, and I thank him and Vida Dujmović for several helpful discussions.

References

[1] Guiseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, 1999. 1

[2] M. J. Carpano. Automatic display of hierarchized graphs for computer aided decision analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, 10(11):705–715, 1980. 1

[3] Sabine Cornelsen, Thomas Schank, and Dorothea Wagner. Drawing graphs on two and three lines. In Michael Goodrich and Stephen Kobourov, editors, *Graph Drawing, 10th International Symposium (GD 2002)*, volume to appear of *Lecture Notes in Computer Science*. Springer-Verlag, 2002. 21

[4] Vida Dujmović, Michael R. Fellows, Michael T. Hallett, M. Kitching, Giuseppe Liotta, Catherine McCartin, Naomi Nishimura, Prabhakar Ragde, Frances A. Rosamond, Matthew Suderman, Sue Whitesides, and David R. Wood. On the parameterized complexity of layered graph drawing. In Friedhelm Meyer auf der Heide, editor, *Algorithms, 9th European Symposium (ESA 2001)*, volume 2161 of *Lecture Notes in Computer Science*, pages 488–499. Springer-Verlag, 2001. 20

[5] Vida Dujmović, Matthew Suderman, and David R. Wood. Personal communication, 2002. 4, 8

[6] Peter Eades, Brendan McKay, and Nick Wormald. On an edge crossing problem. In *Proceedings of the 9th Australian Computer Science Conference*, pages 327–334. Australian National University, 1986. 21

[7] John A. Ellis, Ivan Hal Sudborough, and Jonathan Turner. The vertex separation and search number of a graph. *Information and Computation*, 113(1):50–79, 1994. 3, 18

[8] Stefan Felsner, Giuseppe Liotta, and Stephen K. Wismath. Straight-line drawings on restricted integer grids in two and three dimensions (extended abstract). In Petra Mutzel, Michael Jünger, and Sebastian Leipert, editors, *Graph Drawing, 9th International Symposium (GD 2001)*, volume 2265 of *Lecture Notes in Computer Science*, pages 328–342. Springer-Verlag, 2001. 1, 2, 3, 6

[9] Stefan Felsner, Giuseppe Liotta, and Stephen K. Wismath. Straight-line drawings on restricted integer grids in two and three dimensions, manuscript. `http://page.inf.fu-berlin.de/~felsner/Paper/rest-grid.ps.gz`, 2002. 17

[10] Ulrich Fößmeier and Michael Kaufmann. Algorithms and area bounds for nonplanar orthogonal drawings. In Giuseppe Di Battista, editor, *Graph Drawing, 5th International Symposium (GD '97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 134–145. Springer-Verlag, 1997. 21

[11] Frank Harary and Allen Schwenk. A new crossing number for bipartite graphs. *Utilitas Mathematica*, 1:203–209, 1972. 21

[12] Michael Kaufmann and Dorothea Wagner. *Drawing Graphs: Methods and Models*, volume 2025 of *Lecture Notes in Computer Science*. Springer-Verlag, 2001. 1

[13] Thomas Lengauer. *Combinatorial Algorithms for Integrated Circuit Layout*. Wiley, 1990. 1

[14] Petra Mutzel. Optimization in leveled graphs. In Panos M. Pardalos and Christodoulos A. Floudas, editors, *Encyclopedia of Optimization*, pages 189–196. Kluwer Academic Publishers, 2001. 1

[15] Petra Scheffler. A linear algorithm for the pathwidth of trees. In Rainer Bodendiek and Rudolf Henn, editors, *Topics in Combinatorics and Graph Theory*, pages 613–620. Physica-Verlag, Heidelberg, 1990. 3

[16] Konstantin Skodinis. Computing optimal linear layouts of trees in linear time. In Mike Paterson, editor, *Algorithms, 8th European Symposium (ESA 2000)*, volume 1879 of *Lecture Notes in Computer Science*, pages 403–414. Springer-Verlag, 2000. 3

[17] Kozo Sugiyama, Shojiro Tagawa, and Mitsuhiko Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(2):109–125, 1981. 1

[18] John N. Warfield. Crossing theory and hierarchy mapping. *IEEE Transactions on Systems, Man, and Cybernetics*, 7(7):502–523, 1977. 1

[19] Michael S. Waterman and Jerrold R. Griggs. Interval graphs and maps of DNA. *Bulletin of Mathematical Biology*, 48(2):189–195, 1986. 1