Optimizing Area and Aspect Ratio in Straight-Line Orthogonal Tree Drawings*

TIMOTHY CHAN**
Center for Geometric Computing
Dept. of Computer Science
Johns Hopkins Univ.
tchan@cs.jhu.edu

MICHAEL T. GOODRICH***
Center for Geometric Computing
Dept. of Computer Science
Johns Hopkins Univ.
goodrich@cs.jhu.edu

S. RAO KOSARAJU†
Center for Geometric Computing
Dept. of Computer Science
Johns Hopkins Univ.
kosaraju@cs.jhu.edu

ROBERTO TAMASSIA‡
Center for Geometric Computing
Dept. of Computer Science
Brown Univ.
rt@cs.brown.edu

Abstract. We investigate the problem of drawing an arbitrary n-node binary tree orthogonally in an integer grid using straight-line edges. We show that one can simultaneously achieve good area bounds while also allowing the aspect ratio to be chosen as being $O(1)$ or sometimes even an arbitrary parameter. In addition, we show that one can also achieve an additional desirable aesthetic criterion, which we call "subtree separation." We investigate both upward and non-upward drawings, achieving area bounds of $O(n \log n)$ and $O(n \log \log n)$, respectively, and we show that, at least in the case of upward drawings, our area bound is optimal to within constant factors.

1 Introduction

Binary trees are, of course, very common structures in many application areas, so obtaining good drawings of binary trees is an important com-

ponent in a wide variety of visualization tasks. Nevertheless, there are a number of interesting issues regarding binary-tree drawings that are still unresolved, including those related to drawings that optimize the easily-motivated aesthetic criterion of using straight line segments to display edges while also optimizing the area and aspect ratio of the drawing.

Optimizing the area of a drawing is important, because a drawing typically needs to be displayed on a medium of limited area and resolution, such as a terminal window on a workstation screen. Formally, we define the *area* of a drawing to be the area of a smallest rectangle enclosing the drawing. Of course, this assumes a reasonable rule for defining the resolution of a drawing, such as that used in *grid* drawings, where all vertices are placed at integer grid points and edges are drawn as polygonal chains that bend only at integer grid points. Additionally, one may wish to restrict the drawing further to be an *orthogonal* drawing, which is a drawing where the polygonal chains representing edges must be composed of only vertical and horizontal segments. When drawn on a rastered device such as a laser printer or computer monitor, such drawings avoid the aliasing effect caused by the "staircased" drawing of edges that are neither vertical nor horizontal.

An optimization parameter that is perhaps equal in importance to area for a drawing, however, is the *aspect ratio* of a drawing's enclosing rectangle, i.e., the ratio of the width and height of the rectangle. A drawing that is, for example, tall and narrow would be difficult to display nicely on a printed page or in a screen window even if the area is reasonably small (although it might fit quite nicely on a cash-register tape). Ideally, the aspect ratio should be a parameter that could be chosen from a large range of values, or, failing that, it should at least be allowed to be that of a "well proportioned" rectangle (e.g., 1, 5/3, 8.5/11, or $(1 + \sqrt{5})/2$).

Another aesthetic criterion that may be desirable in some applications is that a tree drawing be *upward*. That is, that the tree be drawn so that no child is placed higher (in the y-direction) than its parent. This criterion is desirable, for example, if the tree represents an inherently hierarchical relationship, such as the organizational structure of a large business.

1.1   Previous related research

There has been a fair amount of research involving area and aspect ratio tradeoffs of tree drawings (e.g., see the annotated bibliography of Di Battista *et al.* [4]). We summarize the previous bounds for planar *polyline* grid drawings, for example, where edges are drawn as polygonal chains that bend only at integer grid points, in Table 1.

Class	Drawing Type	Area	Aspect Ratio(s)	Source
degree-$O(1)$ rooted tree	upward	$\Theta(n)$	$[1/n^{\alpha}, n^{\alpha}]$	[5]
binary tree	upward orthogonal	$\Theta(n \log \log n)$	$O(n \log \log n / \log^2 n)$	[5]
degree-4 tree	orthogonal	$\Theta(n)$	$O(1)$	[6, 7, 11]
degree-4 tree	leaves-on-hull orthogonal	$\Theta(n \log n)$	$O(1)$	[1]

Table 1. Summary of some area/aspect ratio results for planar polyline grid drawings of trees. We use α to denote an arbitrary constant such that $0 \le \alpha < 1$.

Notice that each of the area bounds for polyline drawings are tight, to within constant factors, even for upward orthogonal drawings. The related issues for straight-line drawings are not as well-understood, however. We summarize relevant previous results for this class of drawings in Table 2.

Class	Drawing Type	Area	Aspect Ratio(s)	Source
rooted tree	upward layered grid	$O(n^2)$	$O(1)$	[8]
rooted tree	upward grid	$O(n \log n)$	$O(n/\log n)$	[2, 9]
rooted tree	strictly upward grid	$\Theta(n \log n)$	$O(n/\log n)$	[2]
complete or Fibonacci tree	strictly upward grid	$\Theta(n)$	$O(1)$	[2, 10]
AVL tree	strictly upward grid	$\Theta(n)$	$[(\log^{\beta} n)/n, n/(\log^{\beta} n)]$	[3]
balanced tree of height $O(\log n)$	upward grid	$O(n \log n)$	$O(n/\log n)$	[2, 9]

Table 2. Summary of previous area/aspect ratio results for planar straight-line grid drawings. We use β to denote an arbitrary constant such that $\beta > 1$.

We are not aware, for example, of any non-trivial previous work on straight-line orthogonal grid drawings of arbitrary binary trees. This seems to be a fairly serious omission, since straight-line edges are easier for the eye to follow than polyline edges, and orthogonal drawings automatically avoid small angles between edges, which can also cause confusion, and they also avoid aliasing edges drawn on a rasterized device.

1.2 Subtree separation

There is, in fact, an additional desirable aesthetic property for drawings of binary trees. We say that a region R in the plane is *rectilinearly convex*

if the intersection of R and any vertical or horizontal line is connected. For any set S of integer grid points and edges, define the *rectilinear convex hull* of S to be the smallest rectilinearly-convex region containing S. Let $T[v]$ denote the subtree of a tree T that is rooted at $v \in T$ and contains all the descendents of v in T, i.e., $T[v]$ is the subtree of T induced by v. If, for any disjoint induced subtrees $T[v]$ and $T[w]$ in a binary tree T, the rectilinear convex hulls of $T[v]$ and $T[w]$ are disjoint in a drawing D of T, then we say that D achieves *subtree separation*. This property is desired for binary tree drawings, because it allows the eye to quickly distinguish between different parts of the tree. It also allows for multi-resolutional renderings of a drawing D, so that, for example, if D has too many nodes to all simultaneously fit in a screen window, then D can be rendered up to the resolution of the screen, with some induced subtrees rendered as filled-in rectilinearly-convex regions. Of course, it might not always be possible to achieve subtree separation while also optimizing for other aesthetic criteria. For example, many of the drawings produced by the algorithms of Garg *et al.* [5] do not achieve subtree separation. But it is certainly desirable to achieve this property whenever possible.

1.3   Our results

In this paper we present a general approach, based upon a simple "recursive winding" paradigm, for drawing arbitrary binary trees in small area with good aspect ratio, for both upward and non-upward straight-line orthogonal drawing criteria. Intuitively, our recursive winding paradigm draws a binary tree T by laying down a small chain of nodes monotonically in the x-direction leading to a distinguished node, v_k, and then "winding" by recursively laying out $T[v']$ and $T[v'']$ in the other direction, where v' and v'' denote the children of v_k. We apply this approach in both upward and non-upward drawing frameworks, and we also show that the area bound on our upward orthogonal straight-line drawings is optimal to within constant factors. Specifically, we establish the following results regarding a planar grid drawing, D, of an arbitrary n-node binary tree, T:

- D can be made to realize an upward orthogonal straight-line drawing of T with $O(n \log n)$ area and $O(1)$ aspect ratio. Moreover, D can be made to achieve subtree separation, and it can be produced in $O(n)$ time.

- There are n-node trees that require $\Omega(n \log n)$ area in any upward orthogonal straight-line drawing that achieves an aspect ratio in the range $[1/n^\alpha, n^\alpha]$, for any fixed α with $0 \le \alpha < 1$.

- D can be made to realize a (non-upward) orthogonal straight-line drawing of T with $O(n \log \log n)$ area and aspect ratio in the range $[(\log n)/n, n/\log n]$. Moreover, this D can be made to achieve subtree separation, and it can be produced in $O(n)$ time.

In Section 2 we describe our method for producing upward orthogonal straight-line tree drawings with $O(1)$ aspect ratio. We give our lower bound showing that our $O(n \log n)$ area bound in this case is optimal in Section 3. In Section 4 we describe how one can produce non-upward orthogonal straight-line tree drawings with arbitrary aspect ratios in area that is $O(n \log \log n)$, and we conclude in Section 5.

2   Upward Orthogonal Straight-Line Drawings

We begin by presenting our recursive winding paradigm and show how it can be applied to prove the following:

Theorem 1. *Given any binary tree T with n nodes, there is a planar upward straight-line orthogonal grid drawing of T with height and width $O(\sqrt{n \log n})$. Such a drawing can be constructed in $O(n)$ time, and it achieves subtree separation.*

Without loss of generality, assume that each internal node has degree 2. Given an internal node v, let *left(v)* and *right(v)* denote the left child and the right child of v respectively. Let $T[v]$ again denote the subtree of T rooted at v, and let $N[v]$ be the number of leaves in $T[v]$. Arrange the tree so that $N[left(v)] \leq N[right(v)]$ at every node v. This preprocessing requires only linear time. We first review the following lemma:
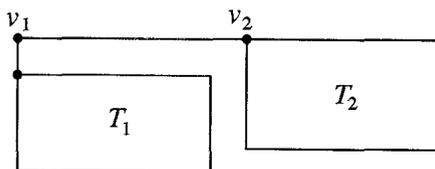

Fig. 1. Drawing of a binary tree with $O(\log n)$ height and $O(n)$ width.

Lemma 2 [2, 9]. *If T has n leaves, then*

(i) *there is a planar upward orthogonal straight-line grid drawing of T with height at most $\lfloor \log_2 n \rfloor$ and width $n-1$, and*

(ii) *there is a planar upward orthogonal straight-line grid drawing of T with width at most $\lfloor \log_2 n \rfloor$ and height $n-1$.*

In both drawings, the root is placed at the upper left-hand corner and the construction time is $O(n)$.

Proof: The construction can be described as follows. If $n = 1$, the drawing is trivial. Suppose $n > 1$ and v_0 is the root of T. Letting $T_1 = T[left(v_0)]$ and $T_2 = T[right(v_0)]$, we can draw T as shown in Figure 1, where the subtrees T_1 and T_2 are drawn recursively. Since $N[left(v_0)] \leq N[right(v_0)]$, it is not difficult to see that the height of the drawing is bounded by $\log_2 n$. Part (ii) can be proven using a similar construction. $\qquad\square$

Before we proceed to prove Theorem 1, we first analyze a certain recurrence relation.

Lemma 3. *Suppose $A > 1$ and f is a function such that*

- *if $n \leq A$, then $f(n) \leq 1$; and*

- *if $n > A$, then $f(n) \leq f(n') + f(n'') + 1$ for some $n', n'' \leq n - A$ with $n' + n'' \leq n$.*

Then $f(n) < 4n/A - 1$ for all $n > A$.

Proof: The proof is by induction. Suppose the theorem is true for n' and n''. If both $n', n'' \leq A$, then $f(n) \leq 3 < 4n/A - 1$. If $n' \leq A$ and $n'' > A$, then

$$f(n) \;\leq\; f(n'') + 2 \;<\; 4n''/A + 1 \;\leq\; 4(n - A)/A + 1 \;<\; 4n/A - 1.$$

Finally, if both $n', n'' > A$, then

$$f(n) \;\leq\; f(n') + f(n'') + 1 \;<\; 4n'/A + 4n''/A - 1 \;\leq\; 4n/A - 1.$$

$\qquad\square$

We now use a recursive winding scheme to prove Theorem 1. Let $H(n)$, $W(n)$, and $T(n)$ denote the height, width, and construction time for drawing a tree with n leaves. Fix a parameter $A > 1$ to be determined later. If $n \leq A$, then we use the scheme in Lemma 2. This provides the base case:

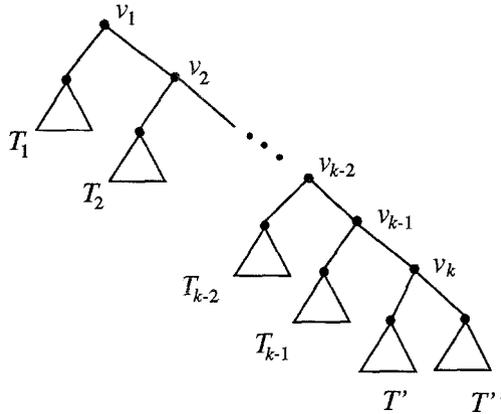$$H(n) \leq \log_2 n, \;\; W(n) \leq A, \;\; \text{and } T(n) = O(A) \;\; \text{if } n \leq A.$$

Fig. 2. The binary tree T.

Suppose $n > A$. Define a sequence $\{v_i\}$ of vertices as follows: v_1 is the root and $v_{i+1} = right(v_i)$ for $i = 1, 2 \ldots$ Let $k \geq 1$ be an index with $N[v_k] > n - A$ and $N[v_{k+1}] \leq n - A$; such an index can be found in $O(k)$ time, since $N[v_1], N[v_2], \ldots$ is a strictly decreasing sequence of integers. Let $T_i = T[left(v_i)]$ and $n_i = N[left(v_i)]$ for $i = 1, \ldots, k - 1$. Let $T' = T[left(v_k)]$, $T'' = T[right(v_k)]$, $n' = N[left(v_k)]$, and $n'' = N[right(v_k)]$. Note that $n' \leq n''$, since T is "right heavy." (See Figure 2.) Note also the following properties:

1. $n_1 + \cdots + n_{k-1} = n - N[v_k] < A$, and

2. $\max\{n', n''\} = N[v_{k+1}] \leq n - A$.

Now, consider the planar upward orthogonal straight-line drawing of T in Figure 3(a), (b), or (c), depending on whether $k = 1$, $k = 2$, or $k > 2$. The root v_1 is placed at the upper left-hand corner. In (a), the subtrees T' and T'' are drawn recursively below v_1. In (b), the subtree T_1 is drawn according to Lemma 2(i), while the subtrees T' and T'' are drawn recursively below this. In (c), the subtrees $T_1, \ldots, T_{k-2}$ are drawn from left to right according to Lemma 2(i); the subtree T_{k-1} is drawn down the right side according to Lemma 2(ii); and the subtrees T' and T'' are drawn recursively below $T_1, \ldots, T_{k-2}$ and then reflected so that their roots are placed at upper right-hand corners (this is the "recursive winding").

In any case, the drawing can be made with the following bounds on the height, width, and construction time:

$$H(n) \leq \max\{H(n') + H(n'') + \log_2 A + 3, \; n_{k-1} - 1\}$$

(a) $k = 1$

(b) $k = 2$

(c) $k > 2$

Fig. 3. Drawing of the binary tree T with $O(\sqrt{n}\log n)$ height and width.

$$W(n) \leq \max\{W(n') + 1,\ W(n''),\ n_1 + \cdots + n_{k-2}\} + \log_2 n_{k-1} + 1$$
$$T(n) \leq T(n') + T(n'') + O(n_1 + \cdots + n_{k-1} + 1).$$

By property 1, we can write the recurrences as

$$H(n) \leq \max\{H(n') + H(n'') + O(\log A),\ A\}$$
$$W(n) \leq \max\{W(n'),\ W(n''),\ A\} + O(\log A)$$

$$T(n) \le T(n') + T(n'') + O(A).$$

By property 2, we can see that $W(n) = O(\lceil n/A \rceil \log A + A)$. Using in addition the fact that $n' + n'' \le n$, we can also conclude that $H(n) = O(\lceil n/A \rceil \log A + A)$ and $T(n) = O(\lceil n/A \rceil A)$ by a direct application of Lemma 3. Theorem 1 then follows by setting $A = \sqrt{n \log_2 n}$.

Moreover, a simple induction argument based upon this proof can be used to show that this construction results in a drawing with subtree separation.

3 A Lower Bound on Upward Orthogonal Straight-Line Drawings

In this section we show that there is a family of n-node trees, for each $n \ge 2$, such that any planar upward straight-line orthogonal grid drawing of such a tree with width W requires $\Omega(n \log n)$ area, for any $\log n \le W \le n/\log n$. We begin with a simple, but important, lemma.

Lemma 4. *If T is an n-node complete binary tree, then any planar straight-line orthogonal grid drawing of T must have width and height at least $\lfloor (\log n)/2 \rfloor$.*

Proof: The proof is based upon an induction argument similar to that used by Crescenzi *et al.* [2] to prove a similar bound for upward (non-orthogonal) grid drawings of T. Let T be an n-node complete binary tree and let D be a straight-line orthogonal drawing of T. We prove that height$(T) \ge (\log n)/2$; the bound for the width follows by a similar argument. Any internal node v in T can have at most two of its adjacent nodes in its same row. If it has one neighbor w in a different row, then the height of the subdrawing for v must be at least one greater than the subdrawing for w, and the subtree of T rooted at w must have size at least half of that of the subtree of T rooted v. On the other hand, if v has two neighbors on the same row as v in D, and one of these nodes, call it w, is not a leaf, and w must have an adjacent node u in T on a different row, and the size of the and the subtree of T rooted at u must have size at least half of that of the subtree of T rooted w. The proof is implied by applying this argument repeatedly (through an induction argument) starting at the root of T. $\qquad\qquad\square$

This lemma is for non-upward orthogonal drawings, but the application we give here is for upward orthogonal drawings.

Theorem 5. *There is an n-node binary tree T that requires $\Omega(n \log n)$ area in any planar straight-line orthogonal upward grid drawing D with width at most n^{α}, for any fixed constant $\alpha > 0$.*

Proof: The proof is based upon a non-trivial adaptation of a "chain pinning" argument of Garg *et al.* [5]. Let $\alpha > 0$ be a fixed constant, and let T be a tree defined by a chain of $n/2$ nodes $C = (v_1, v_2, \ldots, v_{n/2})$ such that every node v_i with i being a multiple of $n^{\alpha/2}$ has a complete binary tree of size $n^{\alpha/2}$ rooted at an adjacent node (not on C). Assume that v_1 is defined as the root of T and partition C into $n^{1-\alpha}/4$ subchains with $2n^{\alpha}$ vertices each. We claim that each subchain S so defined has height that is $\Omega(\log n)$. This claim implies the lemma, since D is upward, so all that remains is to establish this claim.

Let S be a subchain of C as defined above. If the vertical distance between the two endpoints of S is at least $\log n$, then we are done. So, suppose the vertical distance between these two nodes is less than $\log n$. In traversing from endpoint to endpoint of S we alternately march to the left and to the right, possibly with vertical drops mixed into these "left-flowing" and "right-flowing" subchains of S. Since there are less than $\log n$ such chains, there must be two subchains S' and S'' of S that completely cross the same vertical strip of width $2n^{\alpha}/\log n$. Assuming, without loss of generality, that S' is above S'', there are at least $n^{\alpha/2}/\log n > \log n$ subtrees of size $2n^{\alpha}$ each hanging from S'. Let v be a node in S' adjacent to one of these subtrees. Suppose the root of this subtree is on the same vertical level of D as v. Then S' must drop down at least one vertical unit continuing (to the right or left) at this lower level. Since there are at least $\log n$ subtrees like this adjacent to S', this implies that either the subchain S must have height at least $\log n$ or one of these subtrees is positioned directly below the node v on C to which it is adjacent. But if the root of this subtree is directly below v in D, then, by Lemma 4, the subchain S'' must cross the vertical line through v at distance $(\alpha/2) \log n$ below v. This establishes the claim, and the theorem. $\qquad \square$

Having established tight asymptotic bounds on the area of nice-aspect-ratio upward straight-line orthogonal drawings of binary trees, we next consider non-upward drawings of such trees.

4   Non-Upward Drawings

Let us now consider non-upward straight-line orthogonal grid drawings of a binary tree T in small area with arbitrary aspect ratio, again applying

the recursive winding paradigm. Our approach is similar to that of Section 2, in that we select the chain of vertices $(v_1, v_2, \ldots, v_k)$ according to the same rule as in that section. Likewise, if $k \leq 2$, then we recursively lay out T as in that construction (see Figure 3(a) and (b)). If $k > 2$, however, then we use the construction shown in Figure 4. The subtrees $T_1, \ldots, T_{k-1}$ are drawn according to Lemma 2(i), with the drawing of T_{k-1} rotated 180 degrees. As in the previous construction, the subtrees T' and T''' are drawn recursively and then reflected so that their roots are placed at upper right-hand corners. Also, as in the previous construction, the size of T' is at most that of T'', since T is right-heavy.
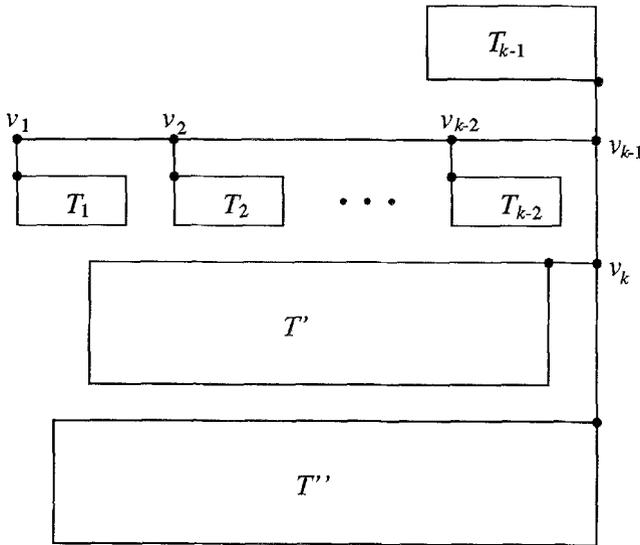

Fig. 4. Recursive strategy for non-upward drawings.

This all implies that the drawing can be made with the following bounds on the height, width, and construction time:

$$H(n) \leq H(n') + H(n'') + 2\log_2 A + 3$$
$$W(n) \leq \max\{W(n') + 1,\ W(n''),\ n_1 + \cdots + n_{k-1}\}$$
$$T(n) \leq T(n') + T(n'') + O(n_1 + \cdots + n_{k-1} + 1).$$

We can re-write the recurrences as

$$H(n) \leq H(n') + H(n'') + O(\log A)$$

$$W(n) \leq \max\{W(n') + 1, \; W(n''), \; A\}$$
$$T(n) \leq T(n') + T(n'') + O(A).$$

Using the fact that $n' \leq n''$, this implies that $W(n) = O(A + \log n)$ and $H(n) = O(\lceil n/A \rceil \log A)$. We can also conclude that $T(n) = O(\lceil n/A \rceil A)$. So, for example, we can take $A = \log n$, and achieve a drawing with width $O(\log n)$ and height $O((n/\log n) \log \log n)$, i.e., the area of this drawing is $O(n \log \log n)$. Of course, the aspect ratio of this drawing is fairly poor.

Still, we can now substitute this method as the base case of our method, rather than using Lemma 2. This allows us to achieve a better aspect ratio while keeping the area to be $O(n \log \log n)$. In particular, this results in the following new recurrence relations characterizing the height and width of the resulting drawing:

$$H(n) \leq H(n') + H(n'') + O(\log A)$$
$$W(n) \leq \max\{W(n') + 1, \; W(n''), \; O((A/\log A) \log \log A)\}.$$

This, in turn, implies that $W(n) = O((A/\log A) \log \log A + \log n)$ while we still have $H(n) = O(\lceil n/A \rceil \log A)$. Thus, we can achieve arbitrary aspect ratio (less than $n/\log n$) while maintaining the area at $O(n \log \log n)$.

This implies the following theorem:

Theorem 6. *Given any binary tree T with n nodes, there is a planar straight-line orthogonal grid drawing of T with area $O(n \log \log n)$. Such a drawing can be constructed in $O(n)$ time so as to achieve subtree separation and any aspect ration in the range $[(\log n)/n, n/\log n]$.*

Proof: As described above, the drawing of T does not necessarily achieve subtree separation. We can easily modify the method, however, to achieve this desired property. The only modification is that in the case $k > 2$ we must be sure that the leftmost point of the drawing of T_{k-1} is to the right of the rightmost point of the drawing of T_{k-2}. Since this modification does not necessitate any changes to the recurrence relations, it achieves the same height and width bounds. □

By Lemma 4 this range of aspect ratios is the largest possible, to within constant factors.

5 Conclusion

We have investigated several issues related to space-efficient planar straight-line orthogonal grid drawings of arbitrary binary trees. In the

case of upward drawings we have established tight upper and lower bounds of $\Theta(n \log n)$ on the area needed, with good aspect ratios, and in the non-upward case we have given a method that achieves area $O(n \log \log n)$ with good aspect ratios. Some interesting problems that remain open include the following:

– Can one achieve arbitrary aspect ratios for planar upward straight-line orthogonal grid drawings while maintaining $O(n \log n)$ area?

– Are there binary trees that require $\Omega(n \log \log n)$ area for any (non-upward) planar straight-line orthogonal grid drawing?

Acknowledgements

References

1. R. P. Brent and H. T. Kung. On the area of binary tree layouts. *Inform. Process. Lett.*, 11:521–534, 1980.
2. P. Crescenzi, G. Di Battista, and A. Piperno. A note on optimal area algorithms for upward drawings of binary trees. *Comput. Geom. Theory Appl.*, 2:187–200, 1992.
3. P. Crescenzi and A. Piperno. Optimal-area upward drawings of AVL trees. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes in Computer Science*, pages 307–317. Springer-Verlag, 1995.
4. G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. Algorithms for drawing graphs: an annotated bibliography. *Comput. Geom. Theory Appl.*, 4:235–282, 1994.
5. A. Garg, M. T. Goodrich, and R. Tamassia. Area-efficient upward tree drawings. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 359–368, 1993.
6. C. E. Leiserson. Area-efficient graph layouts (for VLSI). In *Proc. 21st Annu. IEEE Sympos. Found. Comput. Sci.*, pages 270–281, 1980.
7. C. E. Leiserson. *Area-efficient graph layouts (for VLSI)*. ACM Doctoral Dissertation Award Series. MIT Press, Cambridge, MA, 1983.
8. E. Reingold and J. Tilford. Tidier drawing of trees. *IEEE Trans. Softw. Eng.*, SE-7(2):223–228, 1981.
9. Y. Shiloach. *Arrangements of Planar Graphs on the Planar Lattice*. PhD thesis, Weizmann Institute of Science, 1976.
10. L. Trevisan. A note on minimum-area upward drawing of complete and Fibonacci trees. *Information Processing Letters*, 57(5):231–236, 1996.
11. L. Valiant. Universality considerations in VLSI circuits. *IEEE Trans. Comput.*, C-30(2):135–140, 1981.