Upward order-preserving 8-grid-drawings of binary trees

Therese Biedl*

Abstract

This paper concerns upward order-preserving straight-line drawings of binary trees with the additional constraint that all edges must be routed along edges of the 8-grid (i.e., horizontal, vertical, diagonal) or some subset thereof. We give an algorithm that draws n-node trees with width $O(\log^2 n)$, while the previous best result were drawings of width $O(n^{0.48})$. If only some of the grid-lines are allowed to be used, then the algorithm gives (with minor modifications) the same upper bounds for the $\{\diagup, |, \diagdown\}$-grid and the $\{|, \diagdown, \overline{}\}$-grid. On the other hand, in the $\{\diagup, \diagdown, \overline{}\}$-grid sometimes $\Omega(\sqrt{n/\log n})$ width is required.

1 Introduction

There are many algorithms to draw trees, especially rooted trees, because different applications impose different requirements on the drawing. In this paper, the drawing should satisfy the following constraints:

- It is *planar*, i.e., no vertices or edges overlap unless the corresponding graph-elements do.
- It is *straight-line*, i.e., every edge is drawn as a straight-line segment that connects the corresponding vertices.
- It is *strictly-upward*, i.e., parents have larger y-coordinates that children. (For some of the results this is relaxed to *upward drawings* where edges may be horizontal.)
- It is *order-preserving*, i.e., a given left-to-right order of children at each node must be respected in the drawing.

Such drawings were called *ideal drawings* previously [3]. All drawings in this paper must be planar and straight-line, and this will not always be mentioned. Further, vertices must always be placed at grid-points, i.e., with integer coordinates. Any drawing is assumed (after possible translation) to reside within the $[1, W] \times [1, H]$-grid where W and H are the *width* and *height*. *Column* i consists of all grid-points with x-coordinate i; *row* j consists of all grid-points with y-coordinate j. Since the height may have to be $\Omega(n)$ in a strictly-upward drawing, the objective of this paper is to find ideal drawings of binary trees that have small width.

There are many results concerning how to draw rooted trees; see for example [5] for an overview, [2] for some recent results, and Table 1 for the results especially relevant to this paper. This paper focuses on *grid-drawings*, which means that the edges must be drawn along the lines of a grid. This is well-studied for so-called *orthogonal drawings*, where the grid is the rectangular grid (also called the *4-grid*) and hence all edges are horizontal or vertical. Creszenci et al. [4] showed that every binary n-node tree has an upward straight-line 4-grid-drawing in an $O(\log n) \times O(n)$-grid (the drawing need not be order-preserving). For complete binary trees as well as for Fibonacci trees, they achieve an $O(\sqrt{n}) \times O(\sqrt{n})$-grid. For order-preserving drawings, significantly more area may be needed: Frati [6] showed that $\Omega(n)$ width and height is necessary for some binary trees in an upward straight-line 4-grid drawing.

The focus of this paper is the *octagonal grid* or *8-grid* that has horizontal, vertical and diagonal lines in both directions. Drawings in the 8-grid could also be called *ASCII*-drawings, since they could easily be done in ASCII using characters `/ | _ \`. Creszenci et al. [4] argue that their upward 4-grid-drawings can easily be converted into strictly-upward 8-grid-drawings via a downward shear. This preserves the same width and gives asymptotically the same height, hence any binary tree has an (unordered) strictly-upward drawing in an $O(\log n) \times O(n)$-grid. For order-preserving drawings, only much weaker bounds are known. Chan [3] studied ideal drawings of binary trees (not necessarily with edges along the grid). As he points out, the first and second of his four algorithms adapt easily to create ASCII-drawings of binary trees. The width of these depends much on the chosen *spine* (a concept that will be used in Theorem 1 as well); with a suitable choice Chan achieves ideal 8-grid-drawings of width $O(n^{0.48})$ and height $O(n)$.

Results of this paper: In this paper, we show how to create ideal 8-grid-drawings of binary trees. The previous best known bounds here are drawings of width

*David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Ontario N2L 1A2, Canada. biedl@uwaterloo.ca Supported by NSERC. The author would like to thank Timothy Chan and Stephanie Lee for inspiring discussions.

Grid-lines	Upward?	Order-preserving	width upper bound	width lower bound
$\{\vert, \overline{\ \ }\}$	upward	no	$O(\log n)$ [4]	$\Omega(\log n)$ [4]
$\{\vert, \overline{\ \ }\}$	upward	yes	$O(n)$ (folklore)	$\Omega(n)$ [6]
$\{\vert, \diagdown\}$	strictly-upward	no	$O(\log n)$ [4]	$\Omega(\log n)$ [4]
$\{\diagup, \diagdown\}$	strictly upward	yes	$O(n)$ (folklore)	$\Omega(n)$ (Thm.2)
$\{\vert, \diagdown\}$	strictly upward	yes	$O(n)$ (folklore)	$\Omega(n)$ (Thm.2)
$\{\diagup, \vert, \diagdown\}$	strictly upward	yes	$O(n^{0.48})$ [3] $O(\log^2 n)$ (Thm.1)	$\Omega(\log n)$ [4]
$\{\vert, \diagdown, \overline{\ \ }\}$	upward	yes	$O(\log^2 n)$ (Thm.3)	$\Omega(\log n)$ [4]
$\{\diagup, \overline{\ \ }, \diagdown\}$	upward	no	$O(n)$ (folklore)	$\Omega(\sqrt{n/\log n})$ (Thm.4)

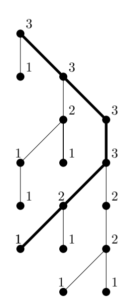
Table 1: Results for planar, upward, straight-line grid-drawings of binary trees. Some more results can be derived in the obvious way, e.g. the upper bound for the $\{\diagup, |, \diagdown\}$-grid also holds for the 8-grid.

$O(n^{0.48})$ [3]. This paper improves this to create drawings that have width $O(\log^2 n)$. In fact, the width is $rpw(T)^2$, where the *rooted pathwidth* $rpw(T)$ is a lower-bound on the width of *any* upward drawing of a tree T (even if it need not be order-preserving or straight-line). Since $rpw(T) \leq \log(n+1)$ [2], our algorithm can be viewed as an $(\log(n+1))$-approximation algorithm for the width of ideal 8-drawings. We also study what happens if one set of the parallel grid-lines is removed; depending on which set is removed we can either achieve the same width-bound or argue that a lower bound of $\Omega(\sqrt{n/\log n})$ holds on the width. See Table 1.

2 Background

Let T be a rooted tree with n nodes that is binary, i.e., every node has at most two children. For any node v, use T_v to denote the subtree rooted at v.

The *rooted pathwidth* $rpw(T)$ of a rooted tree is defined as follows [2]: If every node of T has at most one child, then $rpw(T) = 1$. (In other words, T is a path from the root to the unique leaf.) Otherwise, set $rpw(T) := 1 + \min_P \max_{T' \subset T - P} rpw(T')$. Here the minimum is taken over all paths P for which one end is at the root of T, and the maximum is taken over all subtrees that result when deleting all nodes of P from T. A path P where the minimum is achieved is sometimes called an *rpw-main-path*, though this paper uses the term *spine* to mimic the notations of [7]. The figure above shows a tree with the rooted pathwidth indicated for all nodes; one possible spine is bold.



3 Ideal 8-grid drawings of binary trees

Theorem 1 *Let T be a rooted binary tree. Then T has an ideal 8-grid-drawing of width at most $(rpw(T))^2$.*

Proof. The proof is strongly inspired by the algorithm of Garg and Rusu [7] that give ideal drawings of binary trees of width $O(\log n)$. (Their drawings are not necessarily in the 8-grid.) Their key idea was to use drawings that are "stretchable" in the sense that for any given $\alpha \geq 0$ one can prescribe the contents of the top α rows of the drawing. This then allows to merge drawings of subtrees in a recursive construction. For grid-drawings we need a slightly modified definition as follows:

Definition 1 *Let T be a rooted binary tree with $rpw(T) = r$, and let $\alpha \geq 0$ be given. An 8-grid-drawing of T is called a* left-α-drawing *if within the first α rows, all points in columns $r + 1$ and further to the right are unused.*

Put differently, within the top α rows, only the leftmost r columns may be used for placing vertices and edges. Note that (in contrast to [7]) this definition of a left-α-drawing makes no restrictions where the root must be placed (other than that it must be within the leftmost r columns).[1] Define symmetrically a *right-α-drawing* to be one where within the first α rows only the rightmost r columns may be used.

We need two more types of drawings. Define a *left-corner-drawing* and a *right-corner-drawing* to be a drawing of T where the root is at the top-left (top-right) corner. The main claim, to be proved by induction on $rpw(T)$, is the following:

Claim 1 *Fix an arbitrary $\alpha \geq 0$. Then T has a left-α-drawing, a right-α-drawing, a left-corner-drawing and a right-corner-drawing, and all four drawings have width at most $(rpw(T))^2$.*

To prove this claim, consider the base case where $rpw(T) = 1$. This implies that T is a path from the root to a single leaf. Such a path can easily be drawn

[1] Inspection of the construction given below reveals that the root is always in column 1 or r, but we will not make use of this.

with width $1 = 1^2$, and this satisfies the conditions for all four drawings.

Now assume that $r := rpw(T) \geq 2$. From the definition of rooted pathwidth, we know that T has a spine P such that all subtrees T' of $T - P$ satisfy $rpw(T') \leq rpw(T) - 1$. Let the vertices of P be $v_0 v_1 \ldots v_m$ where v_0 is the root.[2] For simplicity of description, assume that every spine-node $v_i \neq v_0$ has a sibling; if it does not then simply add a sibling and delete it in the obtained drawing later. Adding a sibling that is a leaf does not affect the rooted pathwidth since $rpw(T) \geq 2$, so this does not affect the width-bound. Thus from now on every spine-node except v_m has a left and a right child.

We explain here only how to create the left-α-drawing and the left-corner-drawing; the other two drawings are obtained in a symmetric fashion. There are three cases, depending on whether v_1 is the left or right child of v_0, and which type of drawing is desired.

Case 1: v_1 is the left child of v_0. In this case, the same construction works for both a left-α-drawing and a left-corner drawing (for the latter, use $\alpha := 1$ below). Place the root v_0 at the top-left corner. Let s_0 be the right child of v_0, and recursively obtain a left-α'-drawing $D(T_{s_0})$ of T_{s_0}, where $\alpha' = \alpha - 1$. Place $D(T_{s_0})$, flush left with column 2 and sufficiently far below such that the $\searrow$-diagonal from v_0 ends exactly at s_0. Next obtain recursively a left-corner-drawing $D(T_{v_1})$ of T_{v_1}, and place it below $D(T_{s_0})$, flush left with column 1. Connect (v_0, v_1) vertically (both are in column 1). This finishes the construction.

Note that $rpw(T_{s_0}) \leq rpw(T) - 1 = r - 1$ by definition of rooted pathwidth and the spine. Therefore $D(T_{s_0})$ uses only the leftmost $r-1$ columns within the first α' rows. So this gives a left-α-drawing with the root in the top left corner, as desired. As for the width, $D(T_{s_0})$ has width at most $(r-1)^2$ while $D(T_{v_1})$ has width at most r^2; therefore the width is at most $\max\{1+(r-1)^2, r^2\} = r^2$ as desired.

Case 2: v_1 is the right child of v_0, and we want a left-corner-drawing. Let s_0 be the left child of v_1. Recursively find a left-corner-drawing $D(T_{s_0})$ of T_{s_0}, say that it has height H'. $D(T_{s_0})$ has width at most $(r-1)^2$ since $rpw(T_{s_0}) < rpw(T)$. Recursively find a right-H'-drawing $D(T_{v_1})$ of T_{v_1} of width r^2. (If its width is smaller than r^2, then pad it with empty columns on the left.) Thus within the topmost H' rows of $D(T_{v_1})$, the leftmost $r^2 - r > (r-1)^2$ columns are empty. $D(T_{s_0})$ fits within this empty space; place it flush left with column 1. Finally place v_0 vertically above s_0 (i.e., in column 1) and high enough so that the $\searrow$-diagonal from v_0 ends exactly at v_1. This gives a left-corner-drawing of width r^2 as desired.

[2]The notation here is the same as in [7], though their spine is chosen differently as to always use the heaviest child, rather than the one that has the largest rooted pathwidth.

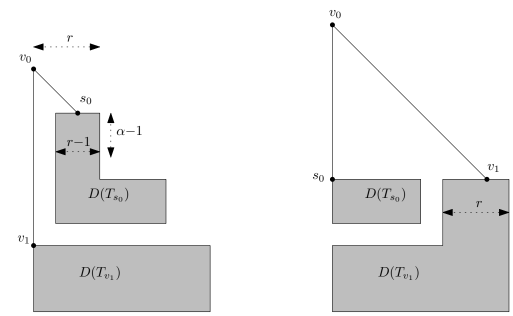


Figure 1: The construction in Case 1 and Case 2.

Case 3: v_1 is the right child of v_0, and we want an α-drawing. This is the most complicated case where a longer section of the spine may get drawn before recursing. Figure 2 illustrates the construction. Recall that every spine-vertex $v_i \neq v_0$ has a sibling by assumption; as in [7] let s_{i-1} be the sibling of v_i. Let $k \geq 1$ be the smallest integer such that v_k is either v_m or s_k is the *left* child of v_k.

First place vertices $v_1, \ldots, v_k$ of the spine; vertex v_0 will be added later. Thus, place v_1 in column r. Now repeat for $1 \leq i \leq k-1$: recursively find a left-corner-drawing $D(T_{s_i})$ of T_{s_i}, place it flush left with column $r+1$ and one row below v_i, then place v_{i+1} in column r and in the last row used by $D(T_{s_i})$. This ends with vertex v_k having been placed in column r. Extend a $\searrow$-diagonal from v_k; this will later be used to complete edge (v_k, v_{k+1}). Next, recursively obtain a left-corner-drawing $D(T_{s_k})$ of T_{s_k}, and place it, flush left with column r and $(r-1)^2$ rows below the row of v_k.

Note that $D(T_{s_k})$ has width at most $(r-1)^2$. Therefore (in the drawing of width r^2 that is being created) there are $r^2 - (r-1)^2 - (r-1) = r$ columns free to the right of $D(T_{s_k})$. These will be used for $T_{v_{k+1}}$ later. Also note that in the topmost row of $D(T_{s_k})$, the $\searrow$-diagonal from v_k is within the rightmost r rows, and hence it does not interfere with $D(T_{s_k})$.

Let H' be the total number of rows that are used thus far, i.e., from the row of v_1 to the bottommost row of $D(T_{s_k})$. Note that columns $1, \ldots, r-1$ are (thus far) entirely free. Recursively find a left-α'-drawing of T_{s_0}, where $\alpha' = H' + \alpha - 1$. Place it, starting $\alpha - 1$ rows above v_1 and flush left with column 1. Within the top α' rows this uses only columns $1, \ldots, r-1$ by $rpw(T_{s_0}) < rpw(T)$, and hence this does not intersect the previously placed subtrees.

Place v_0 vertically above v_1 (i.e., in column r) and high enough so that the $\nearrow$-diagonal from v_0 ends exactly at s_1.

Let H'' be the number of rows from the row of s_k to the bottommost row of $D(T_{s_0})$. Recursively find a right-H''-drawing $D(T_{v_k})$ of T_{v_k}. Place it, flush right with the rightmost column, and in the row of s_k or below such that the $\diagdown$-diagonal extending from v_k exactly meets the point containing v_{k-1}. Within the topmost H'' rows, drawing $D(T_{s_0})$ uses only the rightmost r columns. Recall that r columns remained free next to $D(T_{s_k})$, and also r columns are free next to $D(T_{s_0})$ since this drawing has width at most $(r-1)^2$. Thus drawing $D(T_{v_k})$ does not interfere with previously placed drawings. This gives the desired left-α-drawing of width r^2.

This ends the construction for all cases and proves Theorem 1. □

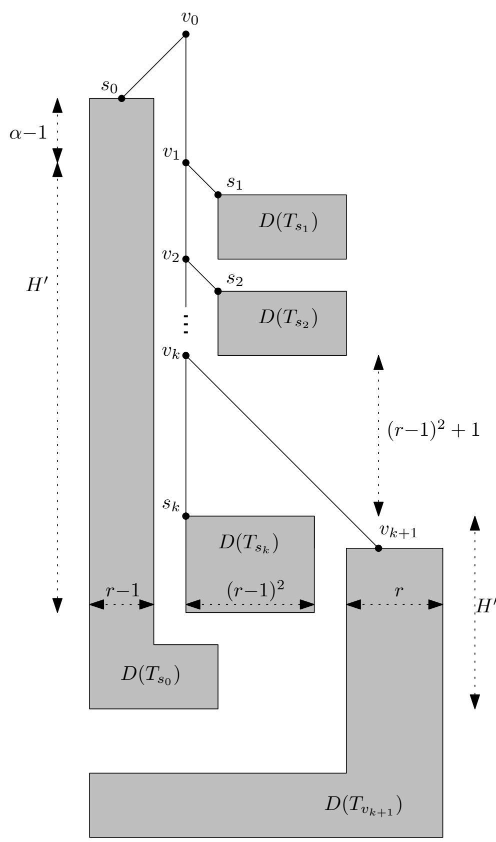


Figure 2: The construction in Case 3.

Height-consideration: In most previous tree-drawing papers, the height is easily shown to be $O(n)$, because all rows (or nearly all rows) intersect at least one vertex. In contrast to this, the construction here has many rows (e.g. most of the rows $1,\dots,r^2$ in the construction for Case 2) that intersect only edges.

One can easily argue that the height is at most $O(n\cdot(rpw(T))^2)$, because (as one can see) any row without vertex in it intersects a diagonal edge, any such diagonal edge intersects at most $rpw(T)^2$ rows, and these rows can be assigned to the upper endpoint of the diagonal edge.

If one follows the construction exactly as described, then $\Omega(nr^2)$ height (for $r = rpw(T)$) may result. (For example, consider a tree where the spine has length $\Omega(n)$ and nearly all siblings of spine-vertices are leaves, but the last few siblings have big enough subtrees to force rooted pathwidth r.) However, there are some obvious possible improvements to the height. To give just one, in Case 2 the drawing $D(T_{s_0})$ could be moved much higher, directly under the $\diagdown$-diagonal, because due to the strict-upwardness of the drawing, the ith row of it is empty in column $i+1$ and farther right. This alone is not enough to ensure a smaller height, but we suspect that combining this with drawing the spine more carefully when some siblings have very small size may lead to a drawing of width $O(\log^2 n)$ and height $O(n)$. This remains for future work.

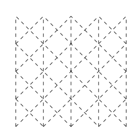
4 Ideal 6-grid drawings of binary trees

Now we turn to the 6-grid, which has grid-lines with angles of 60° between them. Frequently it is easier to think of it instead as a grid that has three of the four sets of grid-lines of the 8-grid (e.g., horizontal, rightward, and $\diagdown$-diagonals). Bachmeier et al. [1] studied 6-grid drawings of trees. Their drawing were not (necessarily) upward, and as such, it was irrelevant which of the grid-lines of the 8-grid are used for the 6-grid, since they are all the same after 90° rotation and/or a shear, and a shear does not affect the asymptotic area.

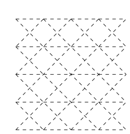
In contrast to this, we study here *upward* drawings of binary trees in the 6-grid, and as before, focus on keeping the width small. As will be seen, here it makes a difference exactly which grid-lines are used to represent the 6-grid.

The following grids will be studied:

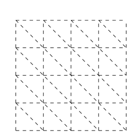
- The $\{\diagup,|,\diagdown\}$-grid: grid-lines are vertical or along a 45° diagonal in either direction.



- The $\{\diagup,\diagdown,\overline{}\}$-grid: grid-lines are horizontal or along a 45° diagonal in either direction.



- The $\{|,\diagdown,\overline{}\}$-grid: grid-lines are horizontal or vertical or along one of the 45° diagonals. (For the $\{\diagup,|,\overline{}\}$-grid a symmetric set of results is obtained by using a horizontal flip.)



We have thus far mostly studied ideal drawings, which must be strictly-upward and hence horizontal lines are disallowed. In particular, Theorem 1 created strictly-upward drawings in the 8-grid, which hence are automatically drawings in the $\{\diagup, |, \diagdown\}$-grid. Therefore every binary tree T has an ideal drawing that is an embedding in the $\{\diagup, |, \diagdown\}$-grid and has width at most $(rpw(T))^2$.

Now we turn to other types of 6-grids. Again, having an ideal drawing means being strictly-upward, so no horizontal lines can be used. We show that then no small width is possible.

Theorem 2 *There exists a binary tree T such that any ideal drawing of T in the $\{|, \diagdown\}$-grid or the $\{\diagup, \diagdown\}$-grid requires width and height $\Omega(n)$.*

Proof. Let T consist of a path of length $n/2$ and attach at each node a left child that is a leaf. For an order-preserving and strictly-upward drawing, the path must be drawn following the $\diagdown$-diagonals. This gives a width and height of at least $n/2 - 1$. □

Therefore, the remaining drawing-results will be in a relaxed model of ideal drawings where horizontal edges are allowed, hence the drawing is upward rather than strictly-upward. Call these *weakly-ideal* drawings. (As before all drawings must be planar, straight-line and order-preserving.)

Theorem 3 *Every binary tree T has a weakly-ideal drawing that is an embedding in the $\{|, \diagdown, \overline{}\}$-grid and has width at most $(rpw(T))^2$.*

Proof. The proof is very similar to the proof of Theorem 1. As before, define (left/right) corner-drawings and (left/right) α-drawings. Additionally now demand for all these drawings that in the topmost row no point to the right of the root is occupied (we say that the root is *right-free*).

Create left-corner-drawings and left-α-drawings almost exactly as before. The only difference is that at the places where a $\diagup$-diagonal was used before, we now use a horizontal edges instead; this is feasible because the root of corresponding subtree is right-free. For right-corner and right-α-drawings, the constructions are not entirely symmetric anymore, but again, by using horizontal edges rather than diagonal ones, drawings can be constructed. Figure 3 illustrates the constructions in all cases; the details are left to the reader. □

Finally consider the $\{\diagup, \diagdown, \overline{}\}$-grid, which is the same as the 8-grid where no vertical edges are allowed. Theorem 2 showed that ideal drawings have to have large width. We show here that even weakly-ideal drawings may require large with. In fact, the following bound holds for any planar straight-line drawing in the $\{\diagup, \diagdown, \overline{}\}$-grid, even if it is not upward or not order-preserving.

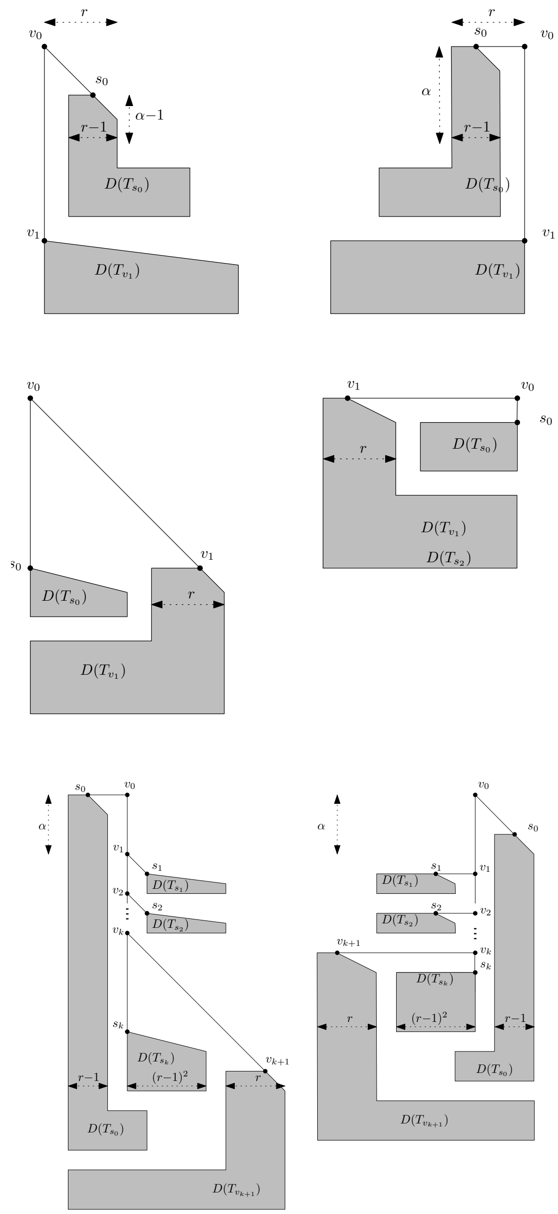


Figure 3: The construction for the $\{|, \diagdown, -\}$-grid.

Theorem 4 *The complete binary tree must have width $O(\sqrt{n/\log n})$ in any straight-line drawing in the $\{\diagup, \diagdown, \overline{}\}$-grid.*

Proof. The proof is very similar to the "simplest" method for obtaining a width-lower-bound for weakly-ideal drawings of the complete ternary tree, see [8]. We repeat the argument here for completeness. Fix an arbitrary straight-line drawing of the complete binary tree in the $\{\diagup, \diagdown, \overline{}\}$-grid, say it has w columns. So any edge spans a horizontal distance of at most $w-1$. Since only horizontal and diagonal edges are allowed, therefore any edge spans a vertical distance of at most $w-1$.

Observe that T has height $h := \log\left(\frac{n+1}{2}\right)$, i.e., the path from the root to each leaf contains h edges. In consequence, any node has vertical distance at most $h(w-1)$ from the root. Therefore the entire drawing is contained within a rectangle that has w columns and up to $h(w-1)$ rows above and below the root, hence $2h(w-1)+1$ rows in total. Therefore the drawing resides in a grid with at most $2wh(w-1)+w$ grid points. Since all n nodes are placed on these grid-points, necessarily

$$n \le 2wh(w-1) + w \in O(w^2 \log n),$$

which implies $w \in \Omega(\sqrt{n/\log n})$. □

We strongly suspect that a lower bound of $\Omega(\sqrt{n})$ on the width holds, but this remains for future work. For the complete binary tree, it is easy to find a construction that has width and height $O(\sqrt{n})$, and in fact, no horizontal edges are used.

Theorem 5 (based on [4]) *The complete binary tree has an ideal drawing in the $\{\diagup, \diagdown\}$-grid of grid-size $O(\sqrt{n}) \times O(\sqrt{n})$.*

Proof. Crescenzi et al. gave a simple recursive construction that draws the complete binary tree in a 4-grid of size $O(\sqrt{n}) \times O(\sqrt{n})$ [4]. Moreover, all edges go rightward or downward. Scale this drawing by $\sqrt{2}$ and then rotate it by 45° clockwise. Due to the scaling, this maps all vertices to grid-points, and all edges are now diagonal and downward as desired. □

5 Remarks

This paper developed algorithms for weakly-ideal 8-grid-drawings of binary trees, i.e., planar upward straight-line order-preserving drawings with edges drawn along grid-lines for the 8-grid (or some subset therefore). We gave constructions of width $O(\log^2 n)$ for a number of such grids. The height is rather large ($O(n \log^2 n)$), and improving this remains an open problem. We also showed that width $O(\sqrt{n/\log n})$ is required for the grid where no vertical lines are allowed.

A natural question is whether similar bounds could be proved for ternary trees. For unordered drawings, Bachmeier et al. [1] gave simple recursive constructions that achieve width $O(n^{\log_3 2}) \approx O(n^{0.631})$. In work done simultaneously with the current paper, Lee studied *ordered* drawings of ternary trees and proved that every ternary tree has such a weakly-ideal 8-grid drawing of width $\Omega(n^{0.68})$ [8]. Furthermore, the complete ternary tree requires width $\Omega(n^{0.411})$ in any upward octagonal-grid-drawing [8]. Both the constructions and the lower bounds in Lee's thesis are significantly more complicated than the ones given here, and will be published separately.

As for open problems, the obvious one is to close the "gap" between the width $O(\log^2 n)$ achieved with our algorithm and the lower bound of $\Omega(\log n)$ for the complete binary tree. Are there binary trees that require $\omega(\log n)$ width in ideal 8-grid drawings?

The other remaining gap concerns drawings in the $\{\diagup, \diagdown, \overline{}\}$-grid-grid. Can we achieve a width of $O(\sqrt{n})$ not just for complete binary trees but for all trees?

References

[1] Christian Bachmaier, Franz-Josef Brandenburg, Wolfgang Brunner, Andreas Hofmeier, Marco Matzeder, and Thomas Unfried. Tree drawings on the hexagonal grid. In Ioannis G. Tollis and Maurizio Patrignani, editors, *Graph Drawing (GD 2008)*, volume 5417 of *Lecture Notes in Computer Science*, pages 372–383. Springer, 2009.

[2] Therese Biedl. Ideal tree-drawings of approximately optimal width (and small height). *Journal of Graph Algorithms and Applications*, 21(4):631–648, 2017.

[3] Timothy M. Chan. A near-linear area bound for drawing binary trees. *Algorithmica*, 34(1):1–13, 2002.

[4] Pierluigi Crescenzi, Giuseppe Di Battista, and Adolfo Piperno. A note on optimal area algorithms for upward drawings of binary trees. *Comput. Geom.*, 2:187–200, 1992.

[5] Giuseppe Di Battista and Fabrizio Frati. A survey on small-area planar graph drawing, 2014. CoRR report 1410.1006.

[6] Fabrizio Frati. Straight-line orthogonal drawings of binary and ternary trees. In Seok-hee Hong, Takao Nishizeki and Wu Quan, editors, *Graph Drawing (GD 2007)*, volume 4875 of *Lecture Notes in Computer Science*, pages 76–87, Springer, 2007.

[7] Ashim Garg and Adrian Rusu. Area-efficient order-preserving planar straight-line drawings of ordered trees. *Int. J. Comput. Geometry Appl.*, 13(6):487–505, 2003.

[8] Stephanie Lee. Upward octagonal drawings of ternary trees. Master's thesis, University of Waterloo, August 2016. (Supervisors: T. Biedl and T. Chan.) Available at `https://uwspace.uwaterloo.ca/handle/10012/10832`.