

Minimum-Area h-v Drawings of Complete Binary Trees

(Extended Abstract)

P. Crescenzi * and P. Penna

Dipartimento di Scienze dell'Informazione
Via Salaria 113, 00198 Roma
e-mail: {piluc, penna}@dsi.uniroma1.it

Abstract. We study the area requirement of h-v drawings of complete binary trees. An h-v drawing of a binary tree t is a drawing of t such that (a) nodes are points with integer coordinates, (b) each edge is either a rightward-horizontal or a downward-vertical straight-line segment from a node to one of its children, (c) edges do not intersect, and (d) if t_1 and t_2 are immediate subtrees of a node u , the enclosing rectangles of the drawings of t_1 and t_2 are disjoint. We prove that, for any complete binary tree t of height $h \geq 3$ and with n nodes, the area of the optimum h-v drawing of t is equal to (a) $2.5n - 4.5\sqrt{(n+1)}/2 + 3.5$ if h is odd, (b) $2.5n - 3.25\sqrt{n+1} + 3.5$ otherwise. As far as we know, this is one of the few examples in which a closed formula for the minimum-area drawing of a graph has been explicitly found. Furthermore this minimum-area h-v drawing can be constructed in linear time. As a consequence of this result and the result of Trevisan (1996), we have that h-v drawings are *provably* less area-efficient than strictly upward drawings when we restrict ourselves to complete binary trees. We also give analogous results for the minimum-perimeter and the minimum-enclosing square area h-v drawings.

1 Introduction

Trees are one of the most common used structures in computer science and many techniques for the visualization of trees have been proposed. These techniques usually aim to find a layout satisfying specific aesthetic criteria. One of these criteria is the strictly upward grid straight-line (in short, *strictly upward*) requirement that imposes to map each node into a point with integer coordinates and each edge into a single straight-line segment, to place each node *below* its parent, and to not intersect two edges. This paper deals with a slight different criteria, that is, the h-v requirement.

* Starting from November 1, 1997, the author's new affiliation will be: Dipartimento di Sistemi ed Informatica, Università di Firenze, Via Cesare Lombroso 6/17, 50134 Firenze, Italy (e-mail: piluc@dsi2.dsi.unifi.it).

The definition of h-v drawing has been introduced in [9] and successively used in [2] as a tool to obtain strictly upward drawings for binary trees. Formally, an *h-v drawing* of a binary tree t is a drawing of t such that (see Fig. 1 and 6):

1. Nodes are points with integer coordinates.
2. Each edge is either a rightward-horizontal or a downward-vertical straight-line segment from a node to one of its children (that is, an h-v drawing is not strictly upward).
3. Edges do not intersect.
4. If t_1 and t_2 are immediate subtrees of a node u , the enclosing rectangles of the drawings of t_1 and t_2 are disjoint.

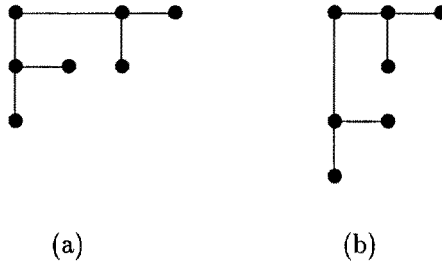


Fig. 1. Two h-v drawings of the complete binary tree of height 3.

The goal of this paper is to find minimum-area h-v drawings of complete binary trees. The width (respectively, height) of an h-v drawing is the width (respectively, height) of the smallest enclosing rectangle². In the following we will denote the height and the width of any h-v drawing Δ with H_Δ and W_Δ , respectively. The *area* of Δ is then defined as $H_\Delta \cdot W_\Delta$.

1.1 Previous Results

In [9] a linear-time algorithm to construct an $O(n \log n)$ -area h-v drawing of any binary tree with n nodes has been proposed. In [2] the authors show that such an algorithm is optimal since an infinite class of binary trees requiring $\Omega(n \log n)$ area exists. Moreover they proved that complete and Fibonacci binary trees admit a linear-area h-v drawing. This latter result has been first extended to AVL trees in [5] and subsequently to k-unbalanced, red-black and BB[α] trees in [3]. A polynomial-time (respectively, efficient parallel) algorithm to find the h-v drawing of any binary tree that minimize any “reasonable size function”

² We adopt the convention that both the width and the height are measured by the number of grid points

is given in [6] (respectively, [8]). In [6] two applications of this algorithm that use general binary trees (not the special case of complete binary trees) to visual interfaces are also presented.

The h-v drawing criterion is a restriction of the orthogonal straight-line one in which each edge of the tree is mapped to either a vertical (not necessarily downward) or an horizontal (not necessarily rightward) straight-line segment. Algorithms to obtain area-efficient orthogonal straight-line drawings of trees can be found in [1, 7, 10].

1.2 Our Results

The algorithm given in [2] for complete binary trees produces an h-v drawing whose area is at most $3n$. It is then natural to ask whether such an area is the minimum one. On the other hand, even if the algorithm in [6] shows that the minimum-area h-v drawing of a complete binary tree can be computed in polynomial time, no closed formula for this area function is known.

In this paper we both show that the algorithm in [2] is not the optimum one and *give a closed formula for the minimum-area requirement to h-v draw a complete binary tree*. In particular, for any such tree t of height $h \geq 3$ and with n nodes, the area of the optimum h-v drawing of t is equal to (a) $2.5n - 4.5\sqrt{(n+1)/2} + 3.5$ if h is odd, (b) $2.5n - 3.25\sqrt{n+1} + 3.5$ otherwise.

This result suggests an interesting comparison with the result of [11]. Indeed, in that paper it is shown that any complete binary tree admits an $(n+o(n))$ -area strictly upward drawing. Our result thus shows that *h-v drawings are provably less area efficient than strictly upward drawings when we restrict ourselves to complete binary trees*. Indeed, it is not difficult to define other less “natural” infinite class of binary trees for which h-v drawings are less area efficient than strictly upward drawings. On the other hand, it is also possible to define an infinite class of binary trees for which strictly upward drawings are less area efficient than h-v drawings (the proof of these two last statements are here omitted: they will be included in the final version of the paper).

Finally, we also observe that analogous results hold for two other size functions defined in [6]: the perimeter and the enclosing-square area.

1.3 Preliminaries

In the following C_h denotes the complete binary tree of height h and n_h denotes the number of nodes of C_h .

For any $h > 1$, given two h-v drawings Δ_1 and Δ_2 of C_{h-1} , we denote by $\Delta_1 \ominus \Delta_2$ and $\Delta_1 \oslash \Delta_2$ the h-v drawings of C_h obtained by combining Δ_1 and Δ_2 as shown in Fig. 2(a) and 2(b), respectively. More precisely, the h-v drawing $\Delta_1 \ominus \Delta_2$ is obtained by translating to the right Δ_2 by as many grid points as W_{Δ_1} and by translating Δ_1 to the bottom by one grid point. The semantic of $\Delta_1 \oslash \Delta_2$ is defined similarly.

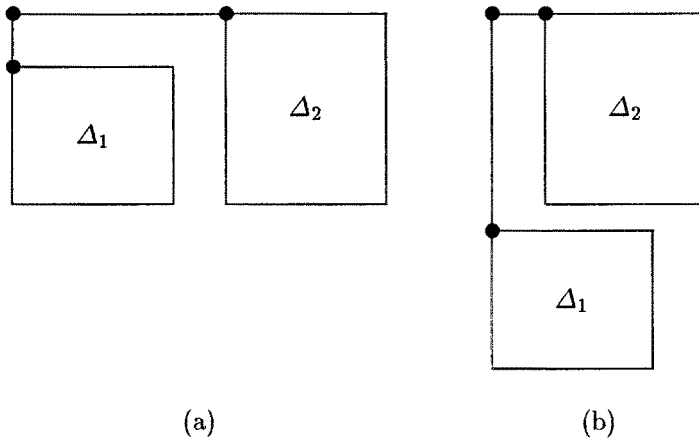


Fig. 2. The two h-v drawing operations.

2 The Algorithm

In this section we describe the algorithm to obtain an h-v drawing for a complete binary tree and we analyze its area requirement. In the next one we prove that this algorithm is optimal.

The algorithm constructs the h-v drawing following a bottom-up approach. More precisely, for each tree C_h , two different h-v drawings are produced: the optimum and the “useful” one. Intuitively, the basic property satisfied by these two drawings is that the width (respectively, height) of the useful drawing is equal to the width (respectively, height) of the optimum drawing minus (respectively, plus) one. On the ground of this property, the two drawings, denoted by O_h and U_h , respectively, are then combined in order to obtain O_{h+1} and U_{h+1} .

Let us first observe that, since we are dealing with complete binary trees, the following fact holds.

Lemma 1. *For any h and for any h-v drawing Δ of C_h , a “reverse” h-v drawing Δ^r of C_h exists such that $W_{\Delta^r} = H_{\Delta}$ and $H_{\Delta^r} = W_{\Delta}$.*

Proof. The proof is by induction on h . If $h = 1$ then we simply let Δ^r be equal to Δ . Otherwise, if $\Delta = \Delta_1 \ominus \Delta_2$ (respectively, $\Delta = \Delta_1 \otimes \Delta_2$) then we define $\Delta^r = \Delta_2^r \otimes \Delta_1^r$ (respectively, $\Delta^r = \Delta_2^r \ominus \Delta_1^r$). \square

The algorithm to construct the two h-v drawings O_h and U_h for C_h (with $h \geq 3$) is described in Fig. 3 where Δ_3 and Δ_3^r denote the h-v drawings shown in Fig. 1(a) and 1(b), respectively.

Theorem 2. *For any $h \geq 3$, the drawing O_h of C_h produced by function MACBT with input h has area equal to*

```

function MACBT(h:integer): h-v drawings  $O_h$  and  $U_h$  of  $C_h$ ;
begin
  if  $h = 3$  then
    MACBT. $O_h := \Delta_3$ ;
    MACBT. $U_h := \Delta_3^r$ ;
  else
    MACBT. $O_h := (\text{MACBT}(h-1).U_{h-1})^r \ominus (\text{MACBT}(h-1).O_{h-1})^r$ ;
    MACBT. $U_h := (\text{MACBT}(h-1).O_{h-1})^r \ominus (\text{MACBT}(h-1).U_{h-1})^r$ ;
  end;

```

Fig. 3. Algorithm to construct O_h and U_h .

$$\begin{cases} 2.5n_h - 4.5\sqrt{(n_h + 1)/2} + 3.5 & \text{if } h \text{ is odd,} \\ 2.5n_h - 3.25\sqrt{n_h + 1} + 3.5 & \text{otherwise.} \end{cases}$$

Proof. The proof is by induction on h . For $h = 3$, the proof is straightforward. Let $h \geq 3$ and let us assume that the theorem is true for any height less than $h + 1$. Let us first observe that, if we denote with L_h and l_h the longer and the shorter side of O_h , respectively, then the longer and the shorter side of U_h is equal to $L_h - 1$ and $l_h + 1$, respectively (see Fig. 4).

Furthermore by the definition of function MACBT we obtain the following recurrence:

$$\begin{cases} L_h = 4 & \text{if } h = 3, \\ l_h = 3 & \text{if } h = 3, \\ L_h = 2l_{h-1} + 1 & \text{if } h > 3, \\ l_h = L_{h-1} & \text{if } h > 3. \end{cases}$$

The solution of this recurrence is:

$$L_h = \begin{cases} 2.5 \cdot 2^{\frac{h-1}{2}} - 1 & \text{if } h \text{ is odd,} \\ 2^{\frac{h+2}{2}} - 1 & \text{if } h \text{ is even,} \end{cases}$$

and

$$l_h = \begin{cases} 2^{\frac{h+1}{2}} - 1 & \text{if } h \text{ is odd,} \\ 2.5 \cdot 2^{\frac{h-2}{2}} - 1 & \text{if } h \text{ is even.} \end{cases}$$

It is then easy to see that the area of O_h , that is $L_h l_h$, is equal to

$$\begin{cases} 2.5n_h - 4.5\sqrt{(n_h + 1)/2} + 3.5 & \text{if } h \text{ is odd,} \\ 2.5n_h - 3.25\sqrt{n_h + 1} + 3.5 & \text{otherwise.} \end{cases}$$

□

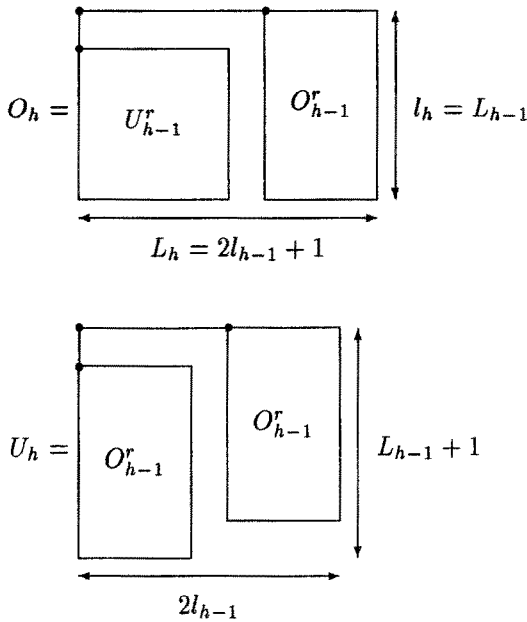


Fig. 4. The composition of O_{h-1} and U_{h-1} to obtain O_h and U_h .

3 The Proof of Optimality

In this section we prove that the h-v drawing O_h is the minimum-area one. To this aim we first need the following definition.

Definition 3 [6]. An h-v drawing Δ for C_h is an *atom* if, for any other h-v drawing Δ' of C_h , either $W_{\Delta'} > W_{\Delta}$ or $H_{\Delta'} > H_{\Delta}$.

It is easy to see that the minimum-area h-v drawing of C_h can be obtained only by combining atoms. We shall therefore suppose in the following that all the drawings are atoms.

Lemma 4. *Let Δ be an h-v drawing for C_h such that $W_{\Delta} \leq H_{\Delta}$. Then an h-v drawing Δ' for the same tree exists such that*

1. $W_{\Delta'} \leq W_{\Delta}$.
2. $\Delta' = \Delta'_1 \otimes \Delta'_2$, for some Δ'_1 and Δ'_2 .
3. The area of Δ' is less than or equal to the area of Δ .

Proof. The proof is by induction on h . For $h = 1$ the theorem is obviously true. Let $\Delta = \Delta_1 \ominus \Delta_2$ (otherwise we can simply define $\Delta' = \Delta$). We distinguish the following two cases.

1. $\Delta_1 = \Delta_2$. From the inductive hypothesis we can assume that $\Delta_1 = \Delta_{1,1} \circ \Delta_{1,2}$, for some $\Delta_{1,1}$ and $\Delta_{1,2}$ (see Fig. 5(a)).
Let Δ' be the h-v drawing defined as (see Fig. 5(b))

$$\Delta' = (\Delta_{1,1} \ominus \Delta_{1,1}) \circ (\Delta_{1,1} \ominus \Delta_{1,2}).$$

If $\Delta_{1,1} = \Delta_{1,2}$, then

$$H_{\Delta'} = H_{\Delta_{1,1}} + 1 + H_{\Delta_{1,1}} + 1 = H_{\Delta_{1,1}} + 1 + H_{\Delta_{1,2}} + 1 = H_{\Delta} + 1$$

and

$$W_{\Delta'} = 2W_{\Delta_{1,1}} + 1 = 2W_{\Delta_{1,2}} + 2 - 1 = W_{\Delta} - 1.$$

Hence the area of Δ' is equal to

$$(H_{\Delta} + 1)(W_{\Delta} - 1) = H_{\Delta}W_{\Delta} + W_{\Delta} - H_{\Delta} - 1 < H_{\Delta}W_{\Delta}$$

where the last inequality is due to the fact that $W_{\Delta} \leq H_{\Delta}$.

Otherwise, if $\Delta_{1,1} \neq \Delta_{1,2}$, it is easy to see that $H_{\Delta_{1,1}} \leq H_{\Delta_{1,2}} - 1$. Indeed, since Δ_1 is an atom we have that $W_{\Delta_{1,1}} > W_{\Delta_{1,2}}$ (otherwise, $\Delta_{1,2} \circ \Delta_{1,1}$ would include Δ_1). From the fact that $\Delta_{1,1}$ and $\Delta_{1,1}$ are atoms it follows that $H_{\Delta_{1,1}} \leq H_{\Delta_{1,2}} - 1$. We then have that

$$H'_{\Delta} = H_{\Delta_{1,1}} + 1 + H_{\Delta_{1,2}} = H_{\Delta}$$

and

$$W_{\Delta'} = 2W_{\Delta_{1,1}} = W_{\Delta}.$$

Thus the area of Δ' is equal to that of Δ .

2. $\Delta_1 \neq \Delta_2$. Observe that, since Δ is an atom we have that $H_{\Delta_2} > H_{\Delta_1}$ (otherwise, $\Delta_2 \ominus \Delta_1$ would include Δ). From the fact that Δ_1 and Δ_2 are atoms it follows that $W_{\Delta_2} < W_{\Delta_1}$. Let us then replace Δ with $\tilde{\Delta} = \Delta_2 \ominus \Delta_2$. The height of $\tilde{\Delta}$ is at most $H_{\Delta} + 1$ while its width is less than W_{Δ} . That is, the area of $\tilde{\Delta}$ is less than or equal to that of Δ . We can now deal with $\tilde{\Delta}$ as in the previous case.

In both the above cases we have shown that the area of Δ' is less than or equal to that of Δ and the lemma thus follows. \square

Informally, the next lemma states that if we shorten the longer side of O_h by c units, then its shorter side increases by at least c units. Conversely, if we shorten its shorter side by c units, then the other side increases by at least $2c$ units. More formally the following fact holds.

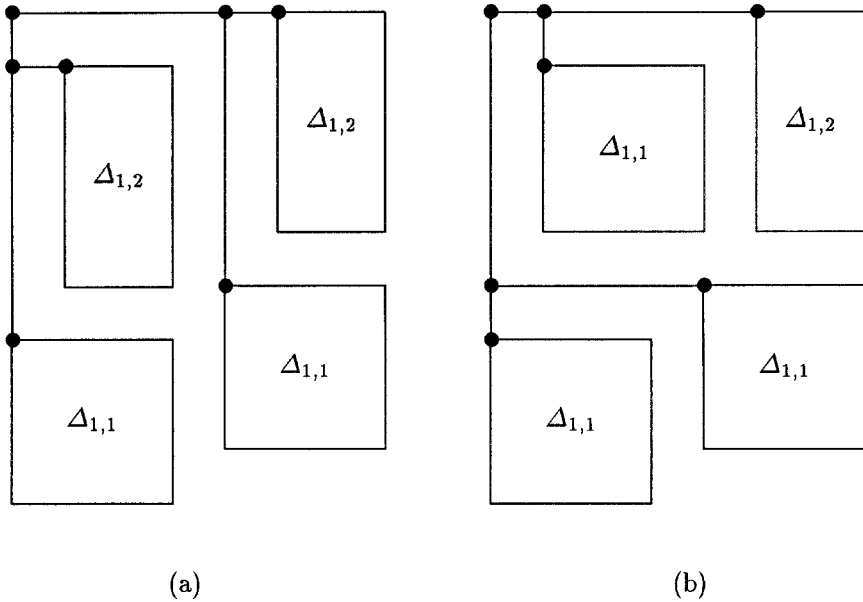


Fig. 5. The transformation of Δ into Δ' .

Lemma 5. For any h-v drawing Δ of C_h (with $h \geq 3$), whose longer and shorter side are L and l , respectively, either

$$(L \leq L_h) \wedge (l \geq l_h + L_h - L)$$

or

$$(l \leq l_h) \wedge (L \geq L_h + 2(l_h - l)).$$

Proof. The proof is by induction on h . For $h \leq 3$ the proof is straightforward. Let us suppose that the lemma holds for any complete tree of height at most h with $h \geq 3$ and let Δ be an h-v drawing for C_{h+1} .

From Lemmas 1 and 4 we can assume, without loss of generality, that the longer side of Δ corresponds to its width and that $\Delta = \Delta_1 \ominus \Delta_2$, for some Δ_1 and Δ_2 .

Since Δ is an atom, either $L = L_{h+1}$ and $l = l_{h+1}$ (in which case the lemma clearly follows) or one of the following two cases must hold.

1. $L < L_{h+1} = 2l_h + 1$. In this case we have that

$$W_{\Delta_1} \leq l_h.$$

Indeed, if $W_{\Delta_1} > l_h$ then $W_{\Delta_2} < l_h$ (since $W_{\Delta_1} + W_{\Delta_2} = L \leq 2l_h$). From the inductive hypothesis it follows that $H_{\Delta_2} \geq L_h + 2(l_h - W_{\Delta_2}) > L_h$. Replacing Δ_1 with O_h yields an h-v drawing for C_{h+1} which is included in

Δ thus contradicting the fact that Δ is an atom. Similarly, we can prove that

$$W_{\Delta_2} \leq l_h.$$

From the inductive hypothesis, it then follows that

$$H_{\Delta_1} \geq L_h + 2(l_h - W_{\Delta_1})$$

and

$$H_{\Delta_2} \geq L_h + 2(l_h - W_{\Delta_2}).$$

Hence

$$\begin{aligned} l &= \max(H_{\Delta_1} + 1, H_{\Delta_2}) \\ &\geq L_h + 2l_h + \max(1 - 2W_{\Delta_1}, -2W_{\Delta_2}) \\ &= 2l_h + L_h - \min(2W_{\Delta_1} - 1, 2W_{\Delta_2}) \\ &\geq 2l_h + L_h + 1 - L = L_{h+1} + l_{h+1} - L, \end{aligned}$$

where the last inequality follows from the fact that $L = W_{\Delta_1} + W_{\Delta_2} \geq \min(2W_{\Delta_1} - 1, 2W_{\Delta_2}) + 1$.

2. $l < l_{h+1}$. In this case, the shorter sides of both Δ_1 and Δ_2 are less than l_h . From the inductive hypothesis, it follows that

$$L^i \geq L_h + 2(l_h - l^i)$$

where L^i and l^i denote the longer and the shorter side of Δ_i , respectively, for $i = 1, 2$. Since $l^i < l_h$, the above inequality implies that

$$L^i + l^i > L_h + l_h.$$

Hence, we have that

$$W_{\Delta_1} > l_h + L_h - H_{\Delta_1}$$

and

$$W_{\Delta_2} > l_h + L_h - H_{\Delta_2}.$$

Since $H_{\Delta_1} + 1 \leq l$ and $H_{\Delta_2} \leq l$, we finally have that

$$L = W_{\Delta_1} + W_{\Delta_2} > 2l_h + 2L_h - 2l + 1 = L_{h+1} + 2(l_{h+1} - l).$$

The lemma thus follows. □

We are now in a position to prove the main result of this section.

Theorem 6. For any $h \geq 3$, O_h is the minimum area h - v drawing for C_h .

Proof. The proof is by induction on h . For $h = 3$ the proof is straightforward. Let $h \geq 3$ and let us assume that the theorem is true for any tree of height less than $h + 1$.

Let Δ be an h-v drawing for C_{h+1} . Let us denote with L and l the longer and the shorter side of Δ , respectively. Since Δ is an atom, either $L = L_{h+1}$ and $l = l_{h+1}$ (in which case the theorem clearly follows) or one of the following two cases must hold.

1. $l < l_{h+1}$.

(a) $l \leq l_h$. From Lemmas 1 and 4, we can assume, without loss of generality, that $W_{\Delta} = L$ and $H_{\Delta} = l$ and that $\Delta = \Delta_1 \ominus \Delta_2$, for some Δ_1 and Δ_2 . Because of the inductive hypothesis, we have that

$$W_{\Delta_1} \geq \frac{L_h l_h}{l-1} > L_h.$$

Thus

$$lL \geq H_{\Delta_1} W_{\Delta_1} + H_{\Delta_2} W_{\Delta_2} + W_{\Delta_1} \geq 2L_h l_h + L_h = L_{h+1} l_{h+1}.$$

(b) $l > l_h$. Because of Lemma 5 we have that

$$L \geq L_{h+1} + 2l_{h+1} - 2l.$$

Thus

$$lL \geq l(L_{h+1} + 2l_{h+1} - 2l) = L_{h+1} l_{h+1} + (l_{h+1} - l)(2l - L_{h+1}) > L_{h+1} l_{h+1},$$

where the last inequality follows from the fact that $l \geq l_h + 1 > L_{h+1}/2$.

2. $L < L_{h+1}$. From Lemma 5 it follows that

$$l \geq l_{h+1} + L_{h+1} - L > l_{h+1}.$$

Thus

$$lL \geq (l_{h+1} + L_{h+1} - L)L = l_{h+1} L_{h+1} + (L_{h+1} - L)(L - l_{h+1}) > l_{h+1} L_{h+1}.$$

Hence, we have proved that the area of Δ is at least equal to the area of O_{h+1} . The theorem thus follows. \square

In Fig. 6(a) and 6(b) the h-v drawing obtained by the algorithm proposed in [2] and the minimum-area h-v drawing, respectively, for C_5 are shown³. Observe that the area of the first h-v drawing is 70 while the area of the minimum one is 63.

³ An "animation" of our algorithm with input C_5 is available from the WEB home page of the first author: <http://www.dsi.uniroma1.it/~piluc/>

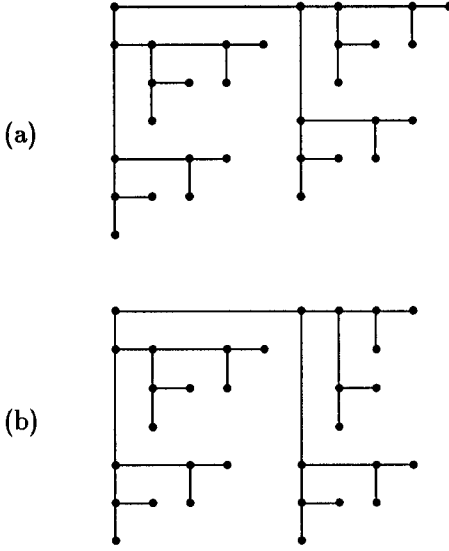


Fig. 6. Two h-v drawings for C_5 .

4 Two More Size Functions

In this section we consider two other size functions among those defined in [6]. In particular, for any h-v drawing Δ , we define

1. $perimeter(\Delta) = H_\Delta + W_\Delta$.
2. $square(\Delta) = \max(H_\Delta, W_\Delta)$.

Let us first observe that from Lemma 5, we have that $perimeter(\Delta) \geq l_h + L_h$, for any drawing Δ of C_h . It thus follows that O_h also minimizes the $perimeter(\cdot)$ function.

Let us now consider the $square(\cdot)$ function. Again using Lemma 5 we have that, for any h-v drawing Δ of C_h , $square(\Delta) \geq \frac{l_h + L_h}{2}$. In order to find the h-v drawing for which $square(\cdot)$ is minimized, we define

$$S_h = \begin{cases} U_h & \text{if } h \leq 5, \\ (U_{h-2} \ominus U_{h-2}) \otimes (U_{h-2} \oplus U_{h-2}^r) & \text{if } h = 6, \\ (S_{h-2} \ominus S_{h-2}) \otimes (\bar{U}_{h-2} \oplus S_{h-2}) & \text{otherwise,} \end{cases}$$

where

$$\bar{U}_h = \begin{cases} O_h & \text{if } h \leq 5, \\ (O_{h-2} \ominus U_{h-2}) \otimes (U_{h-2} \oplus U_{h-2}) & \text{if } h = 6, \\ (\bar{U}_{h-2} \ominus S_{h-2}) \otimes (\bar{U}_{h-2} \oplus S_{h-2}) & \text{otherwise.} \end{cases}$$

It is easy to see that, for any h with $3 \leq h \leq 5$, S_h is the minimum enclosing square drawing. Moreover, for $h \geq 6$, we have that its longer side is equal to $\frac{l_h + L_h}{2}$, which implies that S_h is the minimum-enclosing square area h-v drawing. It is interesting to note that the constant factor of S_h is different depending on the fact that h is odd or even.

5 Open Questions

The first problem left open by this paper is that of obtaining similar results for other balanced trees (e.g. Fibonacci trees). Indeed, we conjecture that the algorithm to construct h-v drawings of Fibonacci trees presented in [2] computes the minimum-area h-v drawings of such trees.

Secondly, the upper bound known for AVL trees, whose class includes both complete and Fibonacci trees, is $18n$ [4], that is, much more than the lower bound we proved in this paper. We think that for this class, the gap between the upper and the lower bound could be reduced by focusing on a particular kind of AVL tree, requiring more area than any other AVL tree with the same height.

Finally, it may also be interesting to find a linear-time algorithm computing the minimum area h-v drawing of any binary tree. Observe that, for the complete and Fibonacci binary trees, this fact already holds.

References

1. T. Chan, M. T. Goodrich, S. R. Kosaraju, and R. Tamassia. Optimizing Area and Aspect Ratio in Straight-Line Orthogonal Tree Drawings. In *Proc. Graph Drawing 96*, LNCS 1190, 63-75, 1997.
2. P. Crescenzi, G. Di Battista, and A. Piperno. A note on optimal area algorithms for upward drawings of binary trees. *Computational Geometry: Theory and Applications*, 2:187-200, 1992.
3. P. Crescenzi and P. Penna. Upward Drawings of Search Trees. In *Proc. WG 96*, LNCS 1197, 114-125, 1997.
4. P. Crescenzi, P. Penna, and A. Piperno. Linear area upward drawings of AVL trees. *Computational Geometry: Theory and Applications*, to appear.
5. P. Crescenzi, and A. Piperno. Optimal-area upward drawings of AVL trees. In *Proc. Graph Drawing 94*, LNCS 894, 307-317, 1994.
6. P. Eades, T. Lin, and X. Lin. Minimum size h-v drawings. In *Advanced Visual Interfaces*, 386-394, World Scientific, 1992.
7. A. Garg, M. T. Goodrich, and R. Tamassia. Planar upward tree drawings with optimal area. *IJCGA*, 6:333-356, 1996.
8. P.T. Metaxas, G.E. Pantziou, and A. Symvonis. Parallel h-v drawings of Binary Trees. In *Proc ISAAC 94*, 487-496, 1994.
9. Y. Shiloach. Linear and planar arrangements of graphs. Ph.D. Thesis, Department of Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel, 1976.
10. C.-S. Shin, S.K. Kim, and K.-Y. Chwa. Area-Efficient Algorithms for Upward Straight-Line Tree Drawings. In *Proc. COCOON 96*, LNCS 1090, 106-116, 1996.
11. L. Trevisan. A Note on Minimum-Area Upward Drawing of Complete and Fibonacci Trees. *Information Processing Letters*, 57:231-236, 1996.