Information Processing Letters 82 (2002) 237–242

Information Processing Letters

www.elsevier.com/locate/ipl

On the approximability of two tree drawing conventions

Paolo Penna [a,b]

[a] *Dipartimento di Matematica, Università di Roma "Tor Vergata", Via della Ricerca Scientifica, I-00133 Roma, Italy*
[b] *Institut für Teoretische Informatik, ETH Zentrum, Clausiusstrasse 49, CH-8092 Zürich, Switzerland*

Received 7 November 2000; received in revised form 27 July 2001
Communicated by A.A. Bertossi

Abstract

We consider two aesthetic criteria for the visualization of rooted trees: *inclusion* and *tip-over*. Finding the minimum area layout according to either of these two standards is an NP-hard task, even when we restrict ourselves to binary trees.

We provide a fully polynomial time approximation scheme for this problem. This result applies to any tree for tip-over layouts and to bounded degree trees in the case of the inclusion convention. We also prove that such restriction is necessary since, for unbounded degree trees, the inclusion problem is strongly NP-hard. Hence, neither a fully polynomial time approximation scheme nor a pseudopolynomial time algorithm exists, unless P = NP. Our technique, combined with the parallel algorithm by Metaxas et al. [Comput. Geom. 9 (1998) 145–158], also yields an NC fully parallel approximation scheme. This latter result holds for inclusion of binary trees and for the *slicing floorplanning* problem. Although this problem is in P, it is unknown whether it belongs to NC or not. All the above results also apply to other size functions of the drawing (e.g., the perimeter).

Keywords: Approximation algorithms; Tree drawing; Computational complexity; Computational geometry

1. Introduction

Trees are largely used for the representation of hierarchical structures and many techniques and standards for their visualization have been proposed (see [3–7, 10]). In this paper we are interested in *inclusion* and *tip-over* standards [6,4].

In the inclusion convention nodes are represented by boxes, while the parent-child relationship is represented by including one box in another. Moreover, rectangles with the same parent are non-overlapping, one next to the other (same x- or y-coordinate of the top left corner) and within distance at least δ (see Fig. 1). Notice that, for binary trees in which the internal nodes have null size, an optimal area drawing can be obtained by combining the two subdrawings either horizontally or vertically as shown in Fig. 1. The case where internal nodes also contain non-null rectangles can be easily reduced to ternary trees with rectangles

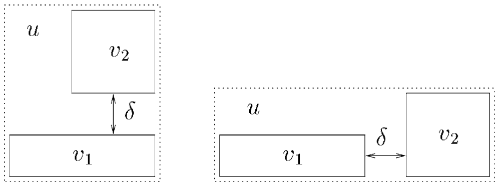

Fig. 1. Inclusion layouts of binary trees with internal nodes with null size: How to arrange the subdrawings.

E-mail address: penna@mat.uniroma2.it (P. Penna).

0020-0190/01/$ – see front matter
PII: S0020-0190(01)00280-0

on the leaves only (see [6] for the details). Notice that in this latter case more than two arrangements of the subdrawings are possible (Fig. 2 shows some of them). More generally, for constant degree trees, only a constant number of arrangements must be considered.

In the tip-over convention, nodes are again represented with boxes, while the children of a node are placed either all horizontally (same x-coordinate of the top left corner), or all vertically (same y-coordinate) as shown in Fig. 3. Notice that in the tip-over convention there are only two possible ways to arrange the drawings of the subtrees: either horizontally or vertically. In the inclusion convention, instead, the layout of a subtree whose root has three or more children can be obtained in several different ways.

Finding the minimum area layout according to one of these two standards is NP-hard, even when we restrict ourselves to binary trees [6]. In the same paper the authors provide a pseudopolynomial time algorithm to compute the optimal solution. It is worth observing (see Theorem 1 below) that such an algorithm guarantees polynomial running time only when the sizes of the rectangles (representing nodes) in the instance are polynomially bounded. Additionally, for the inclusion convention, it applies only to *bounded degree* trees.

In this paper we provide a fully polynomial time approximation scheme (in short FPTAS) for the minimum area layout problem. Our results are based on the algorithm by Eades et al. [6] combined with a rounding technique that allows to achieve, for any error parameter $r > 1$, an r-approximation algorithm running (*regardless of the sizes of the rectangles*) in time $\mathrm{poly}(1/(r-1), n)$.

On the other hand, we show that for unbounded degree trees the minimum area inclusion layout problem is strongly NP-hard. This significantly strengthen the NP-hardness result in [6] in that it implies that nor a fully polynomial time approximation scheme nor a pseudopolynomial time algorithm for that problem exist (see [2,1]). In other words, both the algorithm in [6] and our FPTAS cannot be extended to this case, thus

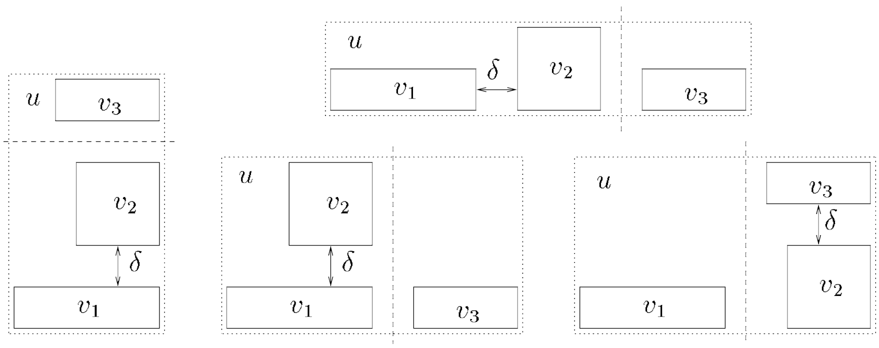

Fig. 2. Some of the possible arrangements for inclusion layouts of ternary trees.

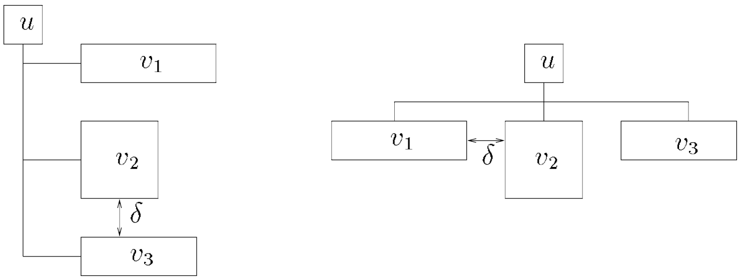

Fig. 3. Tip-over layouts: How to arrange the subdrawings.

Table 1
Hardness and (non-)approximability results for the MIN SIZE INCLUSION and the MIN SIZE TIP-OVER problems

Problem version	Previous results	Our results
MIN SIZE INCLUSION	NP-complete [6]	strongly NP-hard
MIN SIZE TIP-OVER	NP-complete [6]	in FPTAS
MIN SIZE B-INCLUSION	NP-complete [6]	in FPTAS
MIN SIZE 2-INCLUSION	NP-complete [6]	parallel FPTAS
SLICING FLOORPLANNING	in P [11]	parallel FPTAS

making the assumption on the degree of the tree necessary.

We also investigate parallel efficient approximation algorithms. Metaxas et al. in [9] proved that the above mentioned problems are in NC when, again, the sides of the rectangles are polynomially bounded (see Theorem 2 below). We show that our method, combined with such result, gives a fully parallel approximation scheme. The same results also apply to a slightly different problem: *Slicing Floorplanning*. Although this problem is in P [11], the parallel algorithm in [9] does not guarantee a polynomial number of processors when the size of the layout is not polynomially bounded.

Finally, all the results also apply to variants of the problem in which we may want to minimize a function other than the area (namely, the perimeter or the height given a fixed width).

In Table 1 we summarize the hardness and the (non-)approximability results.

1.1. Problem definition and related work

Given a layout $\mathcal{L}$ (satisfying one of the two aesthetic criteria defined above) we denote by $W_{\mathcal{L}}$ and $H_{\mathcal{L}}$ the width and the height of the smallest rectangle enclosing $\mathcal{L}$. We consider the following size functions depending on $W_{\mathcal{L}}$ and $H_{\mathcal{L}}$:

(1) Area: $\text{AREA}(\mathcal{L}) = W_{\mathcal{L}} \cdot H_{\mathcal{L}}$.
(2) Enclosing square area: $\text{SQUARE}(\mathcal{L}) = \max\{W_{\mathcal{L}}, H_{\mathcal{L}}\}^2$.
(3) Perimeter length: $\text{PERIMETER}(\mathcal{L}) = 2(W_{\mathcal{L}} + H_{\mathcal{L}})$.
(4) Height given a fixed width $\overline{w}$:

$$\text{HEIGHT}(\mathcal{L}) = \begin{cases} H_{\mathcal{L}} & \text{if } W_{\mathcal{L}} \leqslant \overline{w}; \\ \infty & \text{otherwise.} \end{cases}$$

(5) Width given a fixed height $\overline{h}$: it is denoted by WIDTH and its definition is symmetric with respect to that of HEIGHT.

The above functions provide a criteria to evaluate the "quality" of a feasible solution (i.e., a layout that satisfies the inclusion or the tip-over aesthetic criteria) depending on the application. Then, depending on the chosen function, we consider the following optimization problems.

MIN AREA INCLUSION.
Instance. A rooted tree T and a function $R : T \to \mathcal{R}^+ \times \mathcal{R}^+$ that maps every node u into a rectangle $R(u)$ whose width and height are given by X_u and Y_u.
Solution. An inclusion layout $\mathcal{L}$.
Measure. The area $\text{AREA}(\mathcal{L}) = W_{\mathcal{L}} \cdot H_{\mathcal{L}}$ of the layout $\mathcal{L}$.

Also, we denote by MIN SIZE INCLUSION the problem defined as MIN AREA INCLUSION where SIZE is *any* of the above size functions. Moreover, we will use MIN WIDTH INCLUSION, MIN PERIMETER INCLUSION, and so on to distinguish a particular size function. All the above definitions will be extended to the TIP-OVER convention. In this case the corresponding optimization problems will be denoted by MIN SIZE TIP-OVER, MIN AREA TIP-OVER, and so on. The MIN SIZE B-INCLUSION problem will be the restriction of MIN SIZE INCLUSION to B-ary trees. A problem strictly related to MIN AREA 2-INCLUSION is SLICING FLOORPLANNING. In this case we are given a regular binary tree in which every leaf is a module (rectangle) and every internal node is labeled either H or V. Moreover, the rectangles representing the leaves can be rotated by 90°, but the arrangement of such boxes is completely determined by the internal nodes (H and V correspond to the vertical and the horizontal arrangement of MIN SIZE 2-INCLUSION). The goal is to minimize the area of the layout.

In the sequel we will make use of the following two results.

Theorem 1 (Eades et al. [6]). *For any positive constant B and for any instance T of* MIN SIZE B-INCLUSION (*or* MIN SIZE TIP-OVER), *the optimum can be computed in* $O(n^2M)$*-time,*[1] *where M is the maximum width* (*or the maximum height*) *of the nodes of T.*

Theorem 2 (Metaxas et al. [9]). *For any instance T of* MIN SIZE 2-INCLUSION (*respectively,* SLICING FLOORPLANNING), *the optimum can be computed in* $O(\log^2 n)$ *parallel time using* $O(n^6(\delta + M)^6/\log n)$ (*respectively,* $O(n^6M^6/\log n)$) *EREW processors, where M is the maximum width/height of the nodes of T.*

Organization of the paper. The rest of the paper is organized as follows: in Section 2 we show how to obtain approximate solutions, while in Section 3 we prove the strong NP-hardness of MIN SIZE INCLUSION. Finally, in Section 4 we consider parallel efficient approximation algorithms and describe some questions left open by this paper.

2. Fully polynomial time approximation scheme

In this section we describe a fully polynomial time approximation scheme that makes use of the pseudopolynomial time algorithm in [6]. We obtain a fully polynomial time approximation scheme by rounding the numbers in the input so that the algorithm of Theorem 1 runs in polynomial time.

Theorem 3. MIN SIZE B-INCLUSION *and* MIN SIZE TIP-OVER *are in* FPTAS.

Proof. For the sake of clarity we first describe the case in which the size function is the area one. Let T be a tree and let X_u and Y_u be the width and the height of any node u of T. We define a truncated instance of the problem as follows. For two suitable integers t_1 and t_2 (we specify their exact value in the sequel), let be

$$X'_u = \left\lfloor \frac{X_u}{10^{t_1}} \right\rfloor \quad \text{and} \quad Y'_u = \left\lfloor \frac{Y_u}{10^{t_2}} \right\rfloor.$$

From the above definition, we have that

$$10^{t_1}X'_u \leqslant X_u < 10^{t_1}X'_u + 10^{t_1} \quad \text{and}$$
$$10^{t_2}Y'_u \leqslant Y_u < 10^{t_2}Y'_u + 10^{t_2}.$$

Let $\mathcal{L}$ be any layout of the tree T and let $W_{\mathcal{L}}$ and $H_{\mathcal{L}}$ denote the width and the height of $\mathcal{L}$, respectively. Let us also denote with $W'_{\mathcal{L}}$ and $H'_{\mathcal{L}}$ the width and the height of $\mathcal{L}$ measured according to the truncated instance. It is then easy to see that

$$10^{t_1}W'_{\mathcal{L}} \leqslant W_{\mathcal{L}} < 10^{t_1}W'_{\mathcal{L}} + 10^{t_1}n, \tag{1}$$
$$10^{t_2}H'_{\mathcal{L}} \leqslant H_{\mathcal{L}} < 10^{t_2}H'_{\mathcal{L}} + 10^{t_2}n. \tag{2}$$

Let us now denote with W_{opt} and H_{opt} the width and the height of the optimal drawing. Also, let $\mathcal{L}$ denote the optimal solution for the truncated instance. From Eqs. (1)–(2) and from the fact that $\mathcal{L}$ is the optimum of the truncated instance, we have that

$$\begin{aligned} &W_{\mathcal{L}}H_{\mathcal{L}} \\ &\quad < \left(10^{t_1}W'_{\mathcal{L}} + 10^{t_1}n\right)\left(10^{t_2}H'_{\mathcal{L}} + 10^{t_2}n\right) \\ &\quad = 10^{t_1+t_2}W'_{\mathcal{L}}H'_{\mathcal{L}} + 10^{t_1+t_2}n\left(H'_{\mathcal{L}} + W'_{\mathcal{L}} + n\right) \\ &\quad \leqslant 10^{t_1+t_2}W'_{\text{opt}}H'_{\text{opt}} + 10^{t_1+t_2}n\left(H'_{\mathcal{L}} + W'_{\mathcal{L}} + n\right) \\ &\quad \leqslant W_{\text{opt}}H_{\text{opt}} + 10^{t_1+t_2}n\left(H'_{\mathcal{L}} + W'_{\mathcal{L}} + n\right). \end{aligned} \tag{3}$$

Thus, if we consider the relative error,[2] Eqs. (1)–(3) imply that

$$\begin{aligned} E &= \frac{W_{\mathcal{L}}H_{\mathcal{L}} - W_{\text{opt}}H_{\text{opt}}}{W_{\mathcal{L}}H_{\mathcal{L}}} \\ &< \frac{10^{t_1+t_2}n(H'_{\mathcal{L}} + W'_{\mathcal{L}} + n)}{W_{\mathcal{L}}H_{\mathcal{L}}} \\ &\leqslant \frac{10^{t_1}n}{W_{\mathcal{L}}} + \frac{10^{t_2}n}{H_{\mathcal{L}}} + \frac{10^{t_1+t_2}n^2}{W_{\mathcal{L}}H_{\mathcal{L}}} \\ &\leqslant \frac{10^{t_1}n}{M_1} + \frac{10^{t_2}n}{M_2} + \frac{10^{t_1+t_2}n^2}{M_1M_2}, \end{aligned}$$

[1] From now on we will denote by n and M, the size and the maximum number occurring in the input set, respectively.

[2] It is easy to see that the relative error corresponds to the approximation ratio minus one.

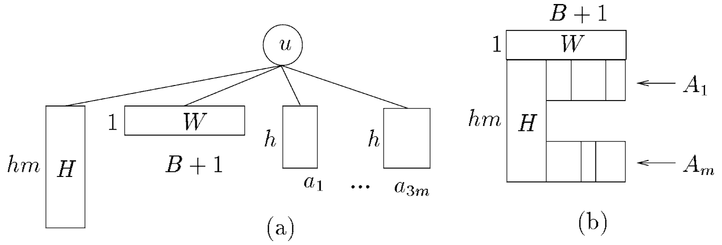

Fig. 4. Strong NP-hardness of MIN SIZE INCLUSION: (a) the MIN SIZE INCLUSION instance corresponding to that of 3-PARTITION; (b) the minimum size layout.

where M_1 and M_2 denote the largest width and height among all the rectangles in the instance, respectively. For any $0 < \varepsilon < 1$, we can make the relative error E smaller than ε by setting t_1 and t_2 as the maximum integers such that

$$\frac{M_1 \cdot \varepsilon}{3 \cdot 10n} < 10^{t_1} \leqslant \frac{M_1 \cdot \varepsilon}{3 \cdot n} \quad \text{and}$$
$$\frac{M_2 \cdot \varepsilon}{3 \cdot 10n} < 10^{t_2} \leqslant \frac{M_2 \cdot \varepsilon}{3 \cdot n}.$$

Hence, $\mathcal{L}$ is a $(1+\varepsilon)$-approximate solution and it can be computed within $O(n^2 \max(M_1/10^{t_1}, M_2/10^{t_2}))$ time, that is, $O(n^3/\varepsilon)$ time.

Finally, it is easy to see that the same method also applies to the PERIMETER function, while it can be easily modified to deal with the HEIGHT and the WIDTH size functions. □

Let us observe that the above result holds in the inclusion convention only for bounded degree trees. In the next section we will prove that such assumption is necessary since, for unbounded degree trees, the problem is strongly NP-hard.

3. Strong NP-hardness of MIN SIZE INCLUSION

In this section we show that, for unbounded degree trees, the MIN SIZE INCLUSION problem is strongly NP-hard. As a consequence we have that neither a pseudopolynomial time algorithm nor a fully polynomial time approximation scheme for such a problem exists. This results holds for *any* of the four size functions defined in Section 1.

Theorem 4. MIN SIZE INCLUSION *is strongly* NP-*hard.*

Proof. We prove the strong NP-hardness of MIN SIZE INCLUSION by reducing 3-PARTITION, which is strongly NP-hard (see [8]), to MIN SIZE INCLUSION.

Let $A = \{a_1, \ldots, a_{3m}\}$ and a positive integer B be an instance of 3-PARTITION. The corresponding instance of MIN SIZE INCLUSION is given by a tree T of height 2 and $3m + 2$ leaves whose rectangles have sides $(1, hm), (B+1, 1), (a_1, h), \ldots, (a_{3m}, h)$, where $h = \lceil (B+1)/m \rceil + 1$ (see Fig. 4(a)).

Without loss of generality, assume $\delta = 0$. Let us observe that A can be partitioned into $3m$ triples of sum B if and only if an inclusion layout $\mathcal{L}^*$ for T such that $W_{\mathcal{L}^*} = B + 1$ and $H_{\mathcal{L}^*} = H + 1$ (see Fig. 4(b)). Finally, from the choice $h = \lceil (B+1)/m \rceil + 1$ it follows that, for any of the size functions defined in Section 1, $\mathcal{L}^*$ is the optimal solution. Indeed, it suffices to observe that any layout whose width and height are $B + 2$ and hm has a non-optimal size. This immediately proves the theorem. □

The following two results are consequence of the previous result and of structural complexity properties of the class FPTAS (see [2,1]).

Corollary 1. MIN SIZE INCLUSION *does not belong to* FPTAS.

Corollary 2. *No pseudopolynomial time algorithm to solve* MIN SIZE INCLUSION *exists.*

4. Extensions and open questions

Concerning efficient parallel algorithms, we observe that our method (see Theorem 3) combined with Theorem 2 yields a parallel FPTAS for MIN SIZE 2-INCLUSION and SLICING FLOORPLANNING. As for the latter, we observe that any feasible solution of SLICING FLOORPLANNING can be regarded as a feasible solution of MIN AREA 2-INCLUSION in which $\delta = 0$, some leaves are rotated and, of course, the arrangement depends on the internal nodes. So, the analysis in Section 2 also applies to this problem. It is worth observing that SLICING FLOORPLANNING can be solved in $O(n^2)$-time [11] while it is not known whether it is in NC or not (the parallel algorithm in [9] only implies that the problem is in NC when the size of the layout is polynomially bounded).

Let us observe that all the results we proved for the minimum area problem also hold for any of the following size functions:

(i) perimeter;
(ii) width given a fixed height;
(iii) height given a fixed width;
(iv) enclosing square.

It seems then natural to try to generalize the results to other size functions. For instance, the results in [6, 9] also hold for *any* function which is monotone in both the height and the width of the drawing. However, it is easy to define some "unnatural" monotone functions for which finding r-approximate solutions is NP-hard, for any arbitrarily large r. It is worth observing that our reduction (working for all the above mentioned functions) subsumes the strongly NP-hardness of 2-DIMENSIONAL PACKING and 2-DIMENSIONAL STRIP PACKING (see [6] for the reduction to MIN AREA INCLUSION).

Finally, it remains open the question whether MIN SIZE INCLUSION can be approximated or not.

Acknowledgements

The author wishes to thank Gianluca Rossi for many useful comments on an earlier version of this paper.

References

[1] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi, Complexity and Approximation—Combinatorial Optimization Problems and their Approximability Properties, Springer, Berlin, 1999.
[2] G. Ausiello, P. Crescenzi, M. Protasi, Appoximate solutions of NP optimization problems, Theoret. Comput. Sci. 150 (1995) 1–55.
[3] P. Crescenzi, G. Di Battista, A. Piperno, A note on optimal area algorithms for upward drawings of binary trees, Comput. Geometry: Theory Appl. 2 (1992) 187–200.
[4] G. Di Battista, P. Eades, R. Tamassia, I. Tollis, Algorithms for drawing graphs: An annotated bibliography, Comput. Geometry: Theory Appl. 4 (1994) 235–282.
[5] G. Di Battista, P. Eades, R. Tamassia, I. Tollis, Graph Drawing—Algorithms for Drawing Graphs, Prentice-Hall, Englewood Cliffs, NJ, 1999.
[6] P. Eades, T. Lin, X. Lin, Two tree drawing conventions, Internat. J. Comput. Geometry Appl. 3 (1996) 133–153.
[7] A. Garg, M.T. Goodrich, R. Tamassia, Planar upward tree drawings with optimal area, Internat. J. Comput. Geometry Appl. 6 (1996) 333–356.
[8] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, New York, 1979.
[9] P.T. Metaxas, G.E. Pantziou, A. Symvonis, A note on parallel algorithms for optimal h-v drawings of binary trees, Comput. Geometry: Theory Appl. 9 (1998) 145–158.
[10] E. Reingold, J. Tilford, Tidier drawings of trees, IEEE Trans. Software Engrg. 2 (1981) 223–228.
[11] L. Stockmeyer, Optimal orientations of cells in slicing floorplan design, Inform. and Control 57 (1993) 91–101.