THE COMPLEXITY OF EMBEDDING GRAPHS INTO BINARY TREES

B. Monien
Universität Paderborn
Warburgerstraße 1oo
479o Paderborn / West Germany

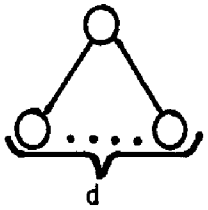
1. Introduction

We consider in this paper the problem of embedding graphs into binary trees. A large variety of problems can be formulated as graph embedding problems where the host graph is a tree. There exists an extensive literature (see [4,5,6,7]). We want to mention here only the problem of representing data structures in the storage of a computer. It is well-known how binary trees are stored and worked with on a computer, therefore the suitability of trees as host structures for various data structures has been of great interest (see [11,12,13,14]). A lot of work also has been done concerning the embedding of graphs into arrays (see [2,8,1o]).As the cost measure of such an embedding we consider in this paper the edge length, i.e. the maximum distance in the binary tree between the images of nodes which are adjacent in G. We give a formal definition below.

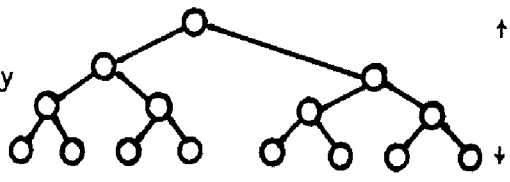
It had been an open problem for some time whether this embedding problem is NP-complete, even for the case of arbitrary guest graphs the complexity was not known. This was contrasting to the behaviour of other host structures, such as lines (bandwidth problem, [3]) or grids, [1]. We show in this paper that the edge length minimization problem for embeddings of graphs into binary trees is NP-complete even when the class of input graphs is restricted to trees of a special structure.

Theorem 1: The edge length minimization problem for embeddings of graphs into binary trees is NP-complete, even when the class of input graphs is restricted to consist only of trees of height 3.

In some sense it is remarkable that this problem is NP-complete already for trees, since for trees there exists an approximation algorithm with a very good behaviour. This (straightforward) algorithm replaces the d sons of a node of the guest tree by the leaves of a subtree of height $\lceil \log_2 d \rceil$ in the host tree, i.e.



is replaced by



It is clear that this algorithm produces an embedding with edge length $\lceil \log_2 \tilde{d} \rceil$ where $\tilde{d}$ is the maximal degree in the input tree. On the other hand it is not difficult to see that for any graph G and for any embedding of G into a binary tree, the edge length is at least $\lceil \log_2 (\text{maxdegree } (G)) \rceil - 2$, see [5]. Therefore the failure of the above algorithm is at most an additive constant of 2. The only other NP-com-

plete problem which we know to have such a good approximation algorithm is the edge colouring problem.

Hong and Rosenberg showed in [5] that outerplanar graphs are almost binary trees, i.e. they showed that for any outerplanar graph G there exists an embedding into a binary tree with edge length $3 \cdot \lceil \log_2 (2 \cdot \text{max degree } (G)) \rceil$. Because of the lower bound mentioned above, this algorithm can be viewed as an approximation algorithm whose failure is at most a multiplicative constant of 3. We wanted to show that outerplanar graphs are even closer related to trees, i.e. we wanted to show that for any outerplanar graph G there exists an embedding into a binary tree with edge length $\lceil \log_2(\text{max degree } (G)) \rceil + c$ for some constant c. Unfortunately we were not able to do so, but we improved the Hong-Rosenberg result considerably.

Theorem 2: Let G be an outerplanar graph. Then there exists an embedding ϵ of G into a binary tree with $\text{cost}(\epsilon) \leq \lceil \log_2 \hat{d} \rceil + \lceil \log_2 \log_2 \hat{d} \rceil + 5$, $\hat{d} = 2 \cdot \text{maxdegree}(G) - 2$.

This algorithm can be viewed again as an approximation algorithm and its worst case performance is bounded by $1 + \epsilon$ for any $\epsilon > 0$.

We have some further results for specially structured graphs. Preorder trees and inorder trees can be embedded into binary trees with edge length 3. This improves results (edge lengths 9 and 6, respectively) from [5]. We could also show that outerplanar graphs of maximal degree 3 can be embedded into binary trees with edge length 3. It is not difficult to see that this result is optimal. For outerplanar graphs of maximal degree 4 the construction in section 3 will show edge length 6. We do not believe that 6 is the optimal result. There are still a lot of open questions to be answered by further research.

We will prove theorem 1 in section 2 and theorem 2 in section 3 and want to give some definitions now.

Let $G = (V,E)$ be a graph. An injective mapping $\epsilon: V \to \{0,1\}^*$ is called an embedding into a binary tree. Note that strings from $\{0,1\}^*$ can be interpreted in a natural way (0 ≡ go to the left, 1 ≡ go to the right) as nodes of a binary tree. Set $\text{Im}(\epsilon) = \{\epsilon(v);\ v \in V\}$. The nodes from $\text{Im}(\epsilon)$ do not necessarily form a tree since we do not demand that for any two nodes $\alpha, \beta \in \text{IM}(\epsilon)$ also all the nodes on the path from α to β belong to $\text{IM}(\epsilon)$. Let us denote by T_ϵ the class of all binary trees whose node set contains $\text{IM}(\epsilon)$, i.e. $T \in T_\epsilon$ iff $T \subset \{0,1\}^*$, T forms a tree, $\text{IM}(\epsilon) \subset T$. The edge length $\text{cost}(\epsilon)$ is defined to be the maximum distance in the binary tree between the images of nodes which are adjacent in G, i.e. $\text{cost}(\epsilon) = \max_{\{u,v\} \in E} \{\text{length of the path between } \epsilon(u) \text{ and } \epsilon(v) \text{ in some tree } T \in T_\epsilon\}$.

We do not consider here embeddings with further restrictions on the image tree. See e.g. [4], where the case is studied that the height of the image tree is as small as possible, i.e. $|\epsilon(v)| < \lceil \log_2(|v| + 1) \rceil - 1$ for all $v \in V$. We want to mention only that our theorem 1 holds also in this case.

2. The complexity of embedding trees

In this section we prove theorem 1.

Theorem 1: The problem of embedding graphs into binary trees is NP complete even when the class of input graphs is restricted to consist only of trees of height 3.

We do the reduction from the 3-partition problem.

input: $q_1,\ldots,q_{3n}$, $p \in \mathbb{N}$

such that

$$p/4 < q_i < p/2 \quad \text{for all } i = 1,..,3n$$

and $\sum_{i=1}^{3n} q_i = n \cdot p.$

problem: does there exist a partition $\{1,\ldots,3n\} = \bigcup_{i=1}^{n} N_i$

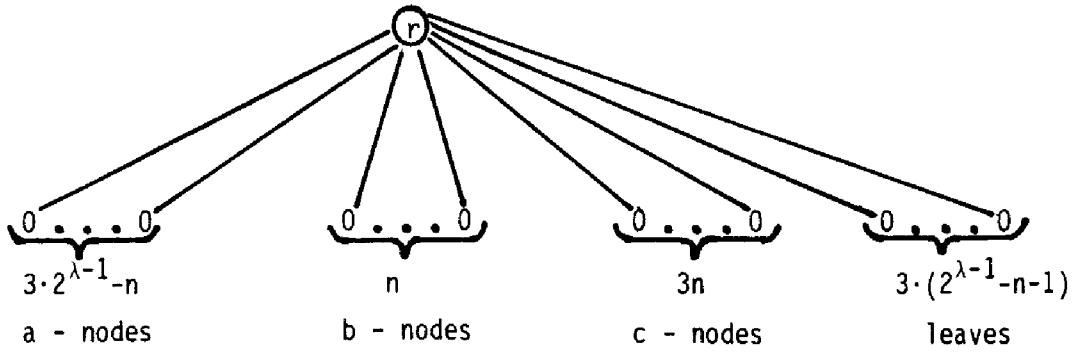
with $|N_i| = 3$ and $\sum_{x \in N_i} x = p$ for all $i = 1,\ldots,n$?

The 3-partition problem is NP complete even in the strong sense, i.e. it remains to be NP complete if the number p is of the same order as the number n. We will consider only the class of problems for which $n \leq p$ holds.

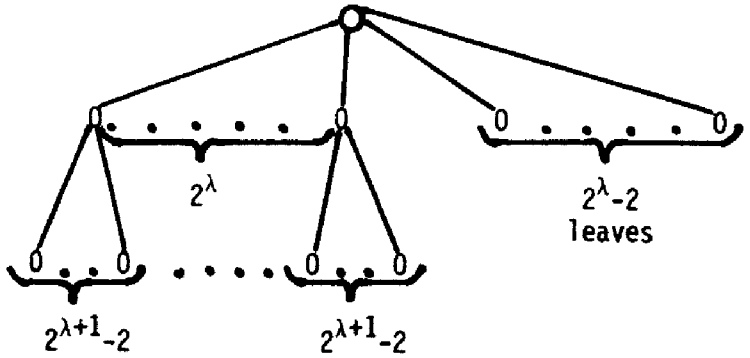
Let μ be the smallest integer such that $p \leq 2^{\mu}-2$ holds. Set $\lambda=\mu+1$. We will construct a tree of height 3 which can be embedded into a binary tree with edge length λ if and only if the 3-partition problem has a solution.

Construction of the tree T

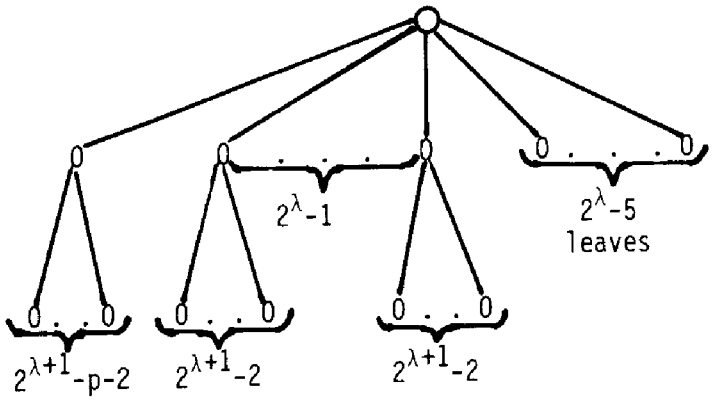
1.) We denote the root of T by r. r has $3\cdot(2^{\lambda}-1)$ sons some of which are leaves. The other sons we distinguish according to the structure of the tree they are rooting as a-,b- and c-nodes, respectively.



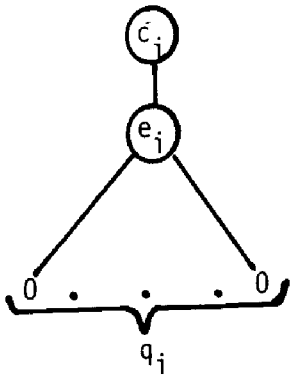
2.) The a-nodes are roots of the following tree.



3.) The b-nodes are roots of the following tree.



4.) Every c-node is the root of a tree encoding on of the numbers q_i.
Denote the corresponding c-node by c_i.

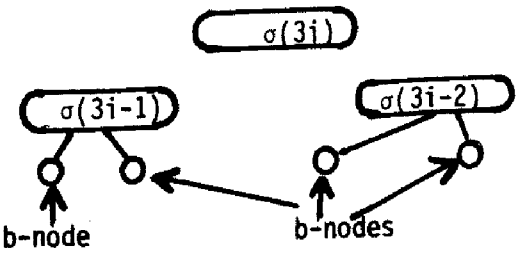


We finish the description of our tree by denoting those sons of a b-node which are no leaves as d-nodes and the distinguished d-nodes as d1-nodes. The above consideration is only possible if $2^{\lambda-1}-n-1 \geq 0$ and $2^{\lambda}-5 \geq 0$ hold. The first condition holds since $n \leq p \leq 2^{\mu}-2$. In order to guarantee the second condition we assume that $\lambda \geq 3$ holds.

Lemma 1: If the 3-partition problem has a solution then T can be embedded into a binary tree with edge length λ.

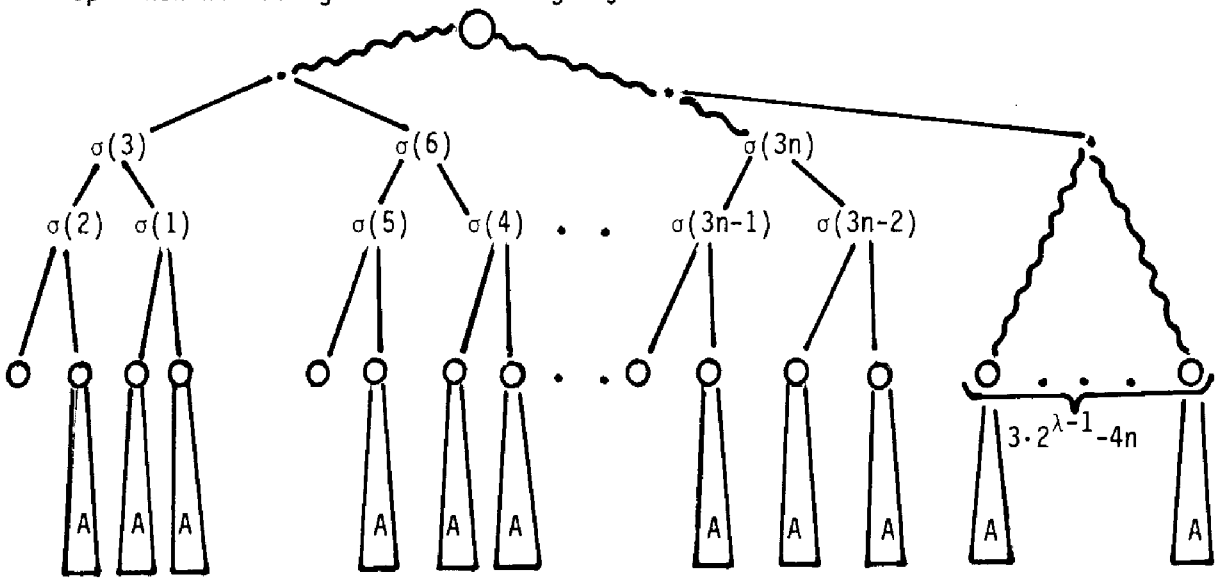
Proof: Let N_i, $i = 1,..,n$, be a solution of the 3-partition problem and let σ: $\{1,..,3n\} \to \{1,..,3n\}$ be a permutation such that $N_i=\{\sigma(3i-2),\sigma(3i-1),\sigma(3i)\}$ holds for all $i = 1,..,n$. Construct a binary tree B in the following way

1.) r is the root of B.
The a-nodes and b-nodes form the level λ of B and the c-nodes and the leaves adjacent to r fill exactly the levels $1,..,\lambda-1$. The c-nodes are placed in such a way that for every i, $1 \leq i \leq n$, the following subgraph is contained in B (here we identify the node c_j with j).

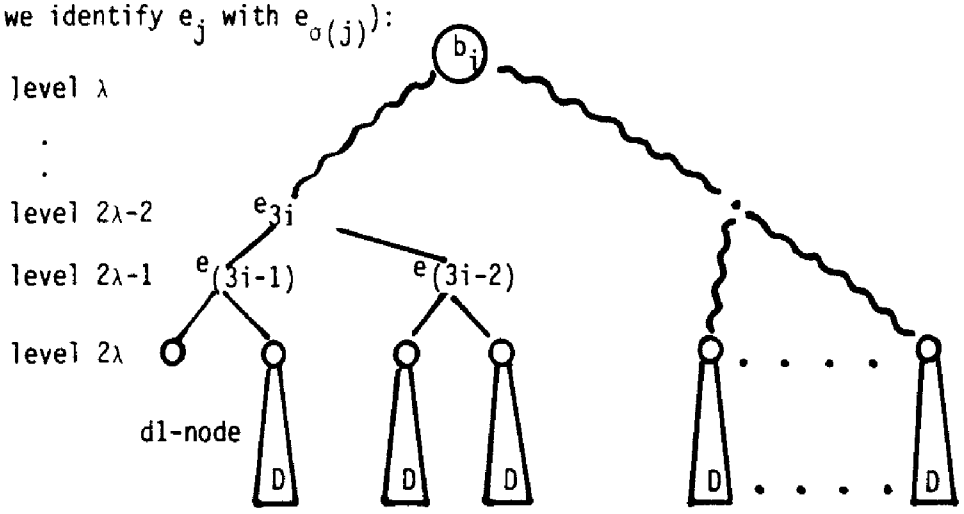


This construction is possible since $n \leq p < 2^{\mu} = 2^{\lambda-1}$ and therefore $3n \leq 3 \cdot 2^{\lambda-1}-p$

2.) The tree rooted at an a-node is laid out as a complete binary tree of height 2λ, with the root at the a-node and with edge length λ. Denote this tree by A. Up to now we have got the following layout.



3.) The sons of a b-node are laid out as a binary tree of depth λ. The d-nodes are laid out in the level 2λ and the leaves adjacent to the b-node are laid out in the levels $\lambda+1,..,2\lambda-1$. The 3 free places are filled with the e-nodes associated already with the corresponding b-node.
If we denote the i-th b-node by b_i then this layout has to have the form (here we identify e_j with $e_{\sigma(j)}$):



The sons of the d-nodes (at this point we still exclude the dl-nodes) are laid out as a complete binary tree of height λ.

4.) Now we have to lay out the $2^{\lambda+1}-p-2$ sons of the dl-node and the $p_{\sigma(3i)}$, $p_{\sigma(3i-1)}$, $p_{\sigma(3i-2)}$ sons of the e-nodes. Note that $P_{\sigma(3i)} + P_{\sigma(3i-1)} + P_{\sigma(3i-2)} = p$ and therefore exactly $2^{\lambda+1}-2$ nodes have to be laid out. We have to show that this can be done with edge length λ.
The sons of the dl-node are laid out as far as possible to the bottom and fill

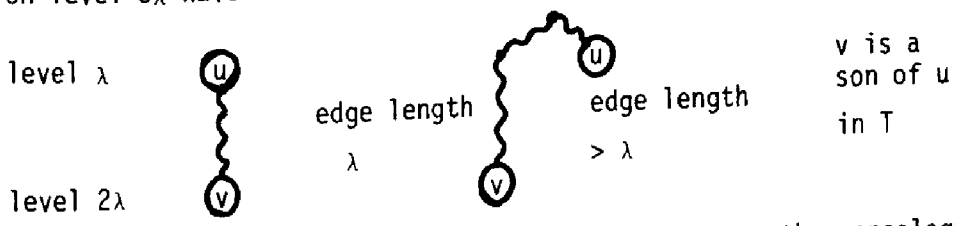
up the levels 3λ and $3\lambda-1$ (this is the true since $2^{\lambda+1}-p-2 \geq 2^{\lambda+1} = 3\cdot 2^{\lambda-1}$). The son of the nodes $e_\sigma(3i)$ and $e_\sigma(3i-1)$ can reach the level $3\lambda-2$ and they fill the rest of this level (this is true since $p_{\sigma(3i-2)} < p/2 \leq 2^{u-1}-1$, i.e. $p_{\sigma(3i-2)} \leq 2^{\lambda-2}-2$). The levels $2\lambda+1,..,2\lambda-3$ can be reached also from node $e_{\sigma(3i-2)}$ with edge length λ and therefore also these levels are filled totally. □

Lemma 2: If T can be embedded into a binary tree with edge length λ then the 3-partition problem has a solution.

Proof: The tree T has $3\cdot(2^\lambda-1)$ sons (i.e. nodes on its level 1), $(3\cdot 2^{\lambda-1}-n)\cdot(2^{\lambda+1}-2) + n\cdot(2^{\lambda+1}-5) + 3n = 3\cdot(2^{\lambda-1}-1)(2^{\lambda+1}-2)$ nodes on its level 2 and $(3\cdot 2^{\lambda-1}-n)\cdot 2\cdot(2^{\lambda+1}-2) + n\cdot(2^{\lambda+1}-p-2) + n\cdot(2^\lambda-1)(2^{\lambda+1}-2) + n\cdot p = (3\cdot 2^{\lambda-1}-1)(2^{\lambda+1}-2)(2^{\lambda+1}-2)$ nodes on its level 3.

Let B be the binary tree into which T has been laid out with edge length λ. The above computation shows that B is a complete binary tree (its root having also degree 3) of depth 3λ and that the levels $i\cdot\lambda+1,\ldots,(i+1)\cdot\lambda$ $(i=0,1)$ are filled exactly with the nodes from T on level i.

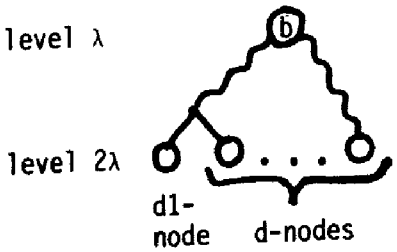
Futhermore the nodes on level 2λ have to be sons of the nodes on level λ and the nodes on level 3λ have to be sons of the nodes on level 2λ.



i.e. at least on the levels λ, 2λ, 3λ the layout preserves the genealogic succession of T (a node on level λ, 2λ, 3λ is the son of the node which is its predecessor of distance λ in T).

As an immediate consequence we get that every node on level λ and level 2λ must have at least 2^λ sons. Therefore the a-nodes and b-nodes form level λ of B and the d-node together with the sons of the a-nodes which are no leaves form the level 2λ of B.

Let us consider now the tree rooted at a b-node chosen arbitrarily. The b-node is laid out on level λ and its successors on level 2λ are exactly the d-nodes adjacent to it.



Note that this structure guarantees also that every grandson of an a-node or b-node, respectively, is laid out in B as a successor of its grandfather. We have seen already that all nodes from T of level 3 have to be laid out in B on level

$2\lambda+1,..,3\lambda$. Therefore the $2^{\lambda}(2^{\lambda+1}-2)$-p grandsons of a b-node are laid out in B on levels $2\lambda+1,..,3\lambda$ as its successors. The free places in these levels can be filled only with the grandsons of some c-nodes. Note that if some grandson of a c-node is a successor of some b-node then all its grandsons are successors of the same b-node. Therefore for every b-node the grandsons of 3 c-nodes are laid out as its successors. Let $\sigma(3i-2)$, $\sigma(3i-1)$, $\sigma(3i)$, $i = 1,..,p$ be the c-nodes whose grandsons are laid out as successors of the i-th b-node. Then $\sigma(3i-2) + \sigma(3i-1) + \sigma(3i) = p$ for all $i = 1,..,p$ and we have found a solution of the 3-partition problem.

□

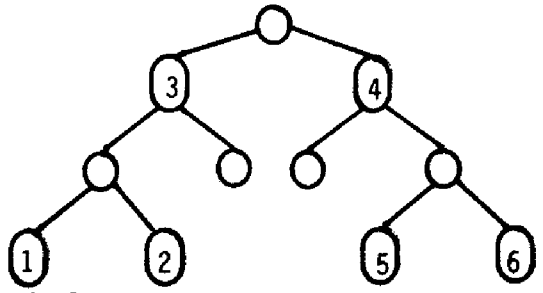
3. Embedding of outerplanar graphs

A graph G is outerplanar if there exists a planar embedding that places all vertices of G in one face. In this section we extend a technique described by J.W. Hong and A.L. Rosenberg in [5]. Hong/Rosenberg showed that if G is an outerplanar graph, then there is an embedding ϵ of G into a binary tree with Cost $(\epsilon) \leq 3\cdot\lceil\log_2(2\text{maxdegree}(G))\rceil$. We improve this result and show that there is an embedding with Cost $\leq \lceil\log_2(\hat{d})\rceil + \lceil\log_2\log_2(\hat{d})\rceil + 5$ where $\hat{d} = 2 \cdot \text{maxdegree}(G) - 2$.

Our proof uses also the technique described in [5]. We look more careful into this technique and show that we can find a good embedding for outerplanar graphs by studying the embedding of a line into a binary tree under a special cost measures. This cost measure is a combination of three wellknown measures. One of this measures is the edge length (denoted by Cost (ϵ)), the other one is the height of the image tree (denoted by height (ϵ)) and the last one (denoted by Dist(ϵ)) describes the distances of the image nodes to the leaves. If all image nodes are leaves then Dist$(\epsilon) = 0$ holds. Otherwise we choose a tree T from T_{ϵ} and we associate to every image node uniquely a leaf from T such that the distance between any image node and the leaf associated to it is as small as poosible, i.e.

$$\text{Dist}(\epsilon) = \min_{T\in T_{\epsilon}} \min_{\substack{\varphi:\text{Im}(\epsilon)\to\text{Leaves}(T)\\ \varphi\text{injective}}} \max_{u\in\text{Im}(\epsilon)} \text{Dist}(u,\varphi(u)).$$

The following figure shows an embedding ϵ of a line of 6 nodes into a binary tree with Cost$(\epsilon) = 2$, Height$(\epsilon) = 3$ and Dist$(\epsilon) = 1$.



We know the optimal embedding of a line with respect to all these measures and we still do so if we combine two of these measures, e.g. there exist wellknown or straight forward embeddings ϵ of a line of n nodes such that

$\text{Cost}(\epsilon) = 3$ and $\text{Height}(\epsilon) = \lceil \log_2(n+1) \rceil - 1$

or $\text{Cost}(\epsilon) = 3$ and $\text{Dist}(\epsilon) = 0$

or $\text{Cost}(\epsilon) = 2$ and $\text{Dist}(\epsilon) = 1$

or $\text{Dist}(\epsilon) = 0$ and $\text{Height}(\epsilon) = \lceil \log_2 n \rceil$.

Here we need a result for the combination of all three measures. We consider $\text{Totalcost}(\epsilon) = \text{Cost}(\epsilon) + \text{Height}(\epsilon) + \text{Dist}(\epsilon)$. We want to find an optimal embedding of a line with respect to this measure, i.e. we are interested in $\text{Line}(n) := \min \{\text{Totalcost}(\epsilon); \epsilon \text{ is the embedding of a line of } n \text{ nodes}\}$.

We defined this measure Totalcost since we can prove the following lemma. The proof is rather technical and is given in [9].

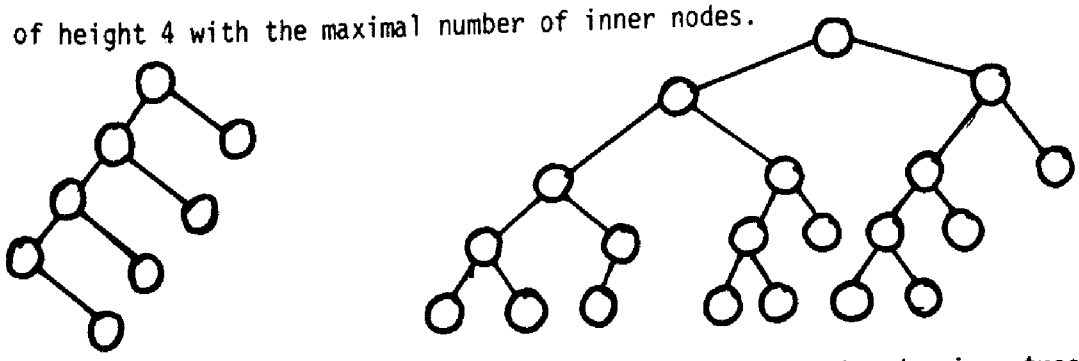
Lemma 3: Every outerplanar graph of maximal degree d can be embedded into a binary tree with edge length $\text{Line}(2d-2)$.

We wanted to show that every outerplanar graph of maximal degree d is embeddable with edge length $\log_2 d + c$ for some fixed constant c. This would be the case if there would exist an embedding of a line which is close to optimal with respect to all three measures. We were not able to show that such a measure exists. In the following we show that $\text{Line}(n) \leq \lceil \log_2 n \rceil + \lceil \log_2 \log_2 n \rceil + 5$ holds. We think that there really exists a trade-off between these three measures.

In order to prove the above result we construct binary trees with the property that to every inner node there is associated a leaf within a fixed distance. Afterwards we use the fact that we can embed a line with edge length 3 into every binary tree with a sufficient number of nodes.

Definition: Let T be a binary tree. Let $V(T) = I \cup L$, where L is the set of leaves of T and I the set of "inner" nodes. A ℓ-mapping on $T, \ell \in \mathbb{N}$, is an injective mapping $\lambda: I \to L$ such that for every $u \in I$ the length of the path from u to $\lambda(u)$ in T is bounded by ℓ. T is called a dist-ℓ-tree if there exists a ℓ-mapping on T.

Note that the only dist-0-tree consists of one node and that the dist-1-trees have the form described in the following figure. This figure shows also the dist-2-tree of height 4 with the maximal number of inner nodes.



The property to be a dist-ℓ-tree influences the number of nodes in a tree of a given height. We let $f_\ell(h)$ be the maximal number of nonleaves of depth h in a dist-ℓ-tree. It is obvious that $f_\ell(h) = 2^h$ for $h < \ell$.

We determine recursively a lower bound for $f_\ell(h)$. We do this explicity for $\ell = 3$. Let T be a dist-3-tree. Then the trees T_1 and T_2, defined by the two sons of the root

of T are also dist-3-trees. Furthermore one of these two trees must have an additiotional leaf of depth ≤ 2 which in the overall construction is associated to the root. Therefore we define additional classes of trees in the following way:

Definition: A binary tree T is called a dist-ℓ-i-tree, $1 \leq i \leq \ell$, if there exists a ℓ-mapping λ on T such that at least one of the leyves of T of depth $\leq i$ is not contained in the domain of λ.

By $f_\ell^i(h)$ we denote the maximal number of nonleavesof depth h in a dist-ℓ-i-tree.

The consideration made above shows that

$f_3(h+1) \geq f_3(h) + f_3^2(h)$ holds.

In order to find a recursive description for $f_3^2(h)$ note that we can construct a dist-3-2-tree by placing the additional leaf of depth ≤ 2 into the left subtree and the leaf associated to the root into the right subtree. Because of this construction we get $f_3^2(h+1) \geq f_3^2(h) + f_3^1(h)$. In a dist-3-1-tree one of the sons of the root has to be a leaf i.e. $f_3^1(h+1) \geq f_2^2(h)$.

Now let g_3, g_3^2, g_3^1 be the functions defined by

$g_3(h) = g_3^2(h) = g_3^1(h) = 2^h$ for $0 \leq h \leq 2$

$g_3(h+1) = g_3(h) + g_3^2(h)$

$g_3^2(h+1) = g_3^2(h) + g_3^1(h)$ for $h \geq 3$

$g_3^1(h+1) = g_3^2(h)$

Then $f_3(h) \geq g_3(h)$ and it is not difficult to transform the recursive equation into $g_3(h+1) = 2g_3(h) - g_3(h-2)$.

In the same way we can show that $f_\ell(h) \geq g_\ell(h)$ for $\ell \geq 1$ where g_ℓ is defined by

$g_\ell(h) = 2^h$ for $h < \ell$

$g_\ell(h) = 2 \cdot g_\ell(h-1) - g_\ell(h-\ell)$ for $h \geq \ell$

We will use this recursive description in order to prove the following lemma:

Lemma 4: $\text{Line}(n) \leq \lceil \log_2 n \rceil + \lceil \log_2 \log_2 n \rceil + 5$

Proof: Using the above recursive description it is not difficult to show that $g_\ell(h) \geq (2 - \frac{4}{3} 2^{1-\ell})^h$ holds for $\ell \geq 4$. Furthermore we constructed dist-ℓ-trees of depth h, $h \in \mathbb{N}$, which have for any i, $0 \leq i \leq h$, at least $g_\ell(i)$ nodes of depth i. Therefore such a tree has

$$\sum_{i=0}^{h} g_\ell(i) \geq \sum_{i=0}^{h} (2 - \frac{4}{3} 2^{1-\ell})^i \geq (2 - \frac{4}{3} 2^{1-\ell})^{h+1} - 1 = \beta(h,\ell) \text{ non-leaves of depth} \leq h.$$

We know that we can embed a line with edge length 3 into every binary tree which has enough nodes. If we use the above construction then we can embed a line of at most $\beta(h,\ell)$ nodes into a binary tree with Cost=3, Dist = ℓ and Height = h. The problem we have to solve now is to choose for any given n the numbers h,ℓ, so that $\beta(h,\ell) \geq n$ holds and $h + \ell$ is as small as possible. We choose $\ell = \lceil \log_2 \log_2 n \rceil + 1$, $h = \lceil \log_2 n \rceil + 1$. Then

$$\beta(h,\ell) + 1 = 2 - \frac{4}{3} \cdot 2^{-\lceil \log_2 \log_2 n \rceil} \lceil \log_2 n \rceil + 2$$

$$\geq 4 \cdot n \cdot (1 - \frac{2}{3 \lceil \log_2 n \rceil}) \lceil \log_2 n \rceil + 2$$

The function $\gamma(r) = 1 - \frac{2}{3r})^{r+2}$ is monotonically increasing in r and $\gamma(5) \geq 0.3$. For $n \geq 32$ we have $\lceil \log_2 n \rceil \geq 5$ and $\lceil \log_2 \log_2 n \rceil + 1 \geq 4$. Therefore $\beta(h,\ell) \geq n$ holds if we choose $h = \lceil \log_2 n \rceil + 1$ and $\ell = \lceil \log_2 \log_2 n \rceil + 1$. I.e. for the case $n \geq 32$ we have found an embedding with Cost = 3, Height = $\lceil \log_2 n \rceil + 1$ and Dist = $\lceil \log_2 \log_2 n \rceil + 1$.

For $n \leq 6$ we have described already an embedding ϵ with Totalcost (ϵ) = 6. It is not difficult to give an embedding fulfilling the lemma for every $n \leq 32$. A complete proof is given in [9]. □

Theorem 2 follows immediatelly from lemma 3 and lemma 4.

Acknowledgement: We want to thank I.H. Sudborough for many helpful discussions. Especially he brought to our attention that the question about the complexity of the Edge Length Minimization Problem for embeddings into binary trees had been an open question for some time.

References

1. Bhatt, S. and S. Cosmadakis; The complexity of minimizing wire lengths in VLSI layouts, manuscript, Lab for Computer Science, MIT, Cambridge, Mass. (1983)

2. R.A. DeMillo, S.C. Eisenstat and R.J. Lipton: Preserving average proximity in arrays, C.ACM 21 (1978), 228 - 231

3. Garey, M.R., R.L. Graham, D.S. Johnson and D.E. Knuth: Complexity results for Badwidth Minimization, SIAM J. Applied Math. 34 (1978), 477 - 495

4. Hong, J.W., K. Mehlhorn and A.L. Rosenberg: Cost Trade-offs in Graph embeddings, with Applications, Journal ACM 3o (1983), 7o9-728

5. Hong, J.W. and A.L. Rosenberg: Graphs that are almost binary trees, SIAM J. Comp. 11 (1982), 227 - 242

6. Paterson, M.S., W.L. Ruzzo and L. Snyder: Bounds on Minimax Edge Length for complete binary trees, ACM Symposium Theory of Computing (1981), 293 - 299

7. D. E. Knuth: The Art of Computer Programming I: Fundamental Algorithms, Addison-Wesley, Reading, MA, 1968

8: R. J. Lipton, S.C. Eisenstat and R.A. DeMillo: Space and time hierarchies for classes of control structures and data structures, J.ACM 23, (1976), 72o - 732

9: B. Monien: The complexity of embedding graphs into binary trees, Technical Report, University of Paderborn, 1985

1o: A.L. Rosenberg: Preserving proximity in arrays, SIAM J. Comput. 4, (1975),443 - 46o

11. A.L. Rosenberg: Data encodings and their costs, Acta Inform.9 (1978), 273 - 292

12. A.L. Rosenberg: Encoding data structures in trees, J. ACM 26 (1979), 668 - 689

13. A. L. Rosenberg and L. Snyder: Bounds on the costs of data encodings, Math. Syst. Th. 12 (1978), 9 - 39

14: A.L. Rosenberg, D. Wood and Z. Galil: Storage representations for tree-like data structures, Math. Syst. Th. 13 (198o), 1o5 - 13o