VERTEX TYPES IN BOOK-EMBEDDINGS

Jonathan F. Buss
Arnold L. Rosenberg
Judson D. Knott

COINS Technical Report 87-18

VERTEX TYPES IN BOOK-EMBEDDINGS

Jonathan F. Buss
Dept. of Computer Science
University of Waterloo
Waterloo, Ontario, CANADA N2L 3G1

Arnold L. Rosenberg
Dept. of Computer and Information Science
University of Massachusetts
Amherst, MA 01003 USA

Judson D. Knott
Dept. of Computer Science
Duke University
Durham, NC 27706 USA

March 6, 1987

Abstract

We study here a new measure of the complexity of a book-embedding of a simple undirected graph, the number of *vertex types* in the embedding. The *type* of a vertex v in a p-page book-embedding is the $p \times 2$ matrix of nonnegative integers

$$\tau(v) = \begin{pmatrix} L_1 & R_1 \\ L_2 & R_2 \\ & \vdots & \\ L_p & R_p \end{pmatrix},$$

where L_i (resp., R_i) is the number of edges incident to v that connect on page i to vertices lying to the left (resp., to the right) of v. The number of types in a book-embedding relates to the amount of logic necessary to realize fault-tolerant arrays of processors using one specific design methodology. We study three sorts of issues regarding vertex types in book-embeddings: We develop a number of techniques for bounding the typenumbers of a variety of graphs. We investigate the relationships between typenumber and other graph properties, such as book thickness. We consider the problem of minimizing the typenumber of a graph. Finding that problem to be *NP*-complete, we study the problem of finding a (cyclic) rotation of a given book-embedding that minimizes the number of vertex types.

1. INTRODUCTION

A *book* is a set of half-planes (the *pages* of the book) that share a common boundary line (the *spine* of the book). An *embedding* of a simple undirected graph[1] G in a book consists

[1] An undirected graph is *simple* if a pair of vertices is connected by at most one edge.

of an ordering of the vertices of G along the spine of the book, together with an assignment of each edge of G to a page of the book, in which edges assigned to the same page do not cross. We shall always assume that a graph is simple and undirected. A ***(cyclic) rotation*** of a given book-embedding is obtained by cyclically rotating the vertices along the spine, while retaining all assignments of edges to pages.

There are three germane measures of the quality of a book-embedding:

1. the ***thickness*** (= number of pages) of the book;
2. the individual and cumulative ***widths of the pages*** (= the cutwidths of the edges on the various pages, and the cutwidth of the entire embedding);
3. the ***number of distinct vertex types***: Given a p-page book-embedding of a graph G, each vertex v of G has an associated $p \times 2$ matrix of nonnegative integers, called its ***type***,

$$\tau(v) = \begin{pmatrix} L_1 & R_1 \\ L_2 & R_2 \\ & \vdots & \\ L_p & R_p \end{pmatrix}. \tag{1}$$

L_i (resp., R_i) is the number of edges incident to v that connect on page i to vertices lying to the left (resp., to the right) of v.

Note that any cyclic permutation (or, ***rotation***) of a p-page book-embedding is another p-page book-embedding.

We henceforth refer to the ***book-thickness*** (also called the ***pagenumber***), the ***pagewidth***, and the ***typenumber*** of a book-embedding and, via minimization over all book-embeddings, of a graph G.

Most work on book-embeddings has focussed solely on book-thickness [1, 2, 5, 8, 10, 15]; some have included pagewidth on their list of concerns [3, 4, 9, 13]. The present paper concentrates on the third cost measure. We study techniques for bounding the typenumbers of graphs by studying a variety of specific graphs (Section 3); we then investigate the relationships between typenumber and a number of other structural properties of graphs (Section 4); we finally consider the problem of minimizing the typenumber of a graph. We show that the problem of finding a typenumber-minimal one-page embedding of an outerplanar graph is NP-complete. We then fall back to the problem of minimizing the typenumber of a given book-embedding, by means of a (cyclic) rotation. We find a rather efficient algorithm for this minimization problem. We also demonstrate the need for such an algorithm by considering the case of complete d-ary trees: the typenumber of any such tree is 3; however, a randomly chosen rotation of the preorder book-embedding of the tree has, with probability approaching 1, $d + 4$ vertex types.

Motivating our study is the DIOGENES methodology for designing fault-tolerant VLSI processor arrays [12, 3]. The methodology views the desired array as an undirected graph, with vertices representing processing elements and edges representing communication links; the design process operates in two stages: First, the graph representing the desired array is embedded in a book; then, the book-embedding is converted to an efficient fault-tolerant layout of the array. The significance of the notion of vertex type is that the type of a vertex "tells it" what role to play in the fault-free processor array. Thus, the base-2 logarithm of the number of vertex types is the number of control bits per processing element needed to configure the array to its fault-free format.

Before turning to the main results of the paper, we present some easily verified basic facts about vertex types and book-embeddings, that are useful in the sequel.

Proposition 1 [1] *The graph G admits a 1-page book-embedding iff G is outerplanar.*

The next result indicates the fundamental nature of vertex types.

Proposition 2 *A book-embedding is determined uniquely by the sequence of vertex types it induces.*

Proof Sketch. Since edges on the same page of a book-embedding cannot cross, one can recreate the entire embedding by reading off (from the sequence of vertex types) how many edges leave the each vertex to the right and to the left on each page of the book and matching these "dangling edges" up in a left-to-right scan of the sequence. For instance, if the sequence of vertices in the book-embedding is

$$v_1, v_2, \cdots, v_n$$

and if on Page k:

- vertex v_i has $e \geq 1$ edges leaving to the right;
- vertex v_j, $j > i$, has at least one edge leaving to the left;
- at most $e - 1$ vertices v_l, $i < l < j$, have edges leaving to the left;
- no vertex v_m, $i < m < j$, has an edge leaving to the right

then we know that vertices v_i and v_j are connected by an edge on Page k. □

The next two results present general upper and lower bounds on the typenumber of an arbitrary graph and of an outerplanar graph.

A nonzero vertex type of the form (1) is a *source* if all $L_i = 0$ and is a *sink* if all $R_i = 0$. A source or a sink is a *one-sided* vertex type; a vertex type that is neither a source nor a sink is *two-sided.*

Proposition 3 *Every graph G having a connected component with at least two vertices has typenumber $t \geq 2$.*

Proof Sketch. Under the stated hypotheses, every book-embedding of G must have a source and a sink. □

We denote by n_G the number of vertices of the graph G and by $n_G(d)$ the number of degree-d vertices of G.

Proposition 4 (a) *In any p-page book-embedding of a graph G, the degree-d vertices of G can assume no more than*

$$\min\left(n_G(d), \binom{2p+d-1}{d}\right) \tag{2}$$

vertex types. Hence, if D_G is the set of all distinct vertex-degrees of the graph G, then the typenumber of the given book-embedding can be no more than

$$\sum_{k\in D_G} \min\left(n_G(k), \binom{2p+k-1}{k}\right). \tag{3}$$

(b) *The typenumber of any book-embedding of G must be at least $|D_G|$.*
(c) *If G is a biconnected outerplanar graph, then the typenumber of any 1-page book-embedding of G cannot exceed*

$$2 + \sum_{k\in D_G} \min\left(n_G(k), k-1\right).$$

Proof Sketch. (a) The two possibilities in Equation (2) hold, respectively, since each vertex has exactly one type, and since each vertex type can be viewed as a partition of the integer d into $2p$ nonnegative parts. Equation (3) is a direct consequence of Equation (2).

(b) The lower bound is immediate since for any degree-d vertex v, $\tau(v)$, as specified in (1), must satisfy $\sum_{i=1}^{p} L_i + \sum_{i=1}^{p} R_i = d$.

(c) A direct instantiation of Equation (3) would yield the bound

$$\sum_{k\in D} \min\left(n_G(k), k+1\right).$$

We can reduce this total by noting that a 1-page book-embedding of a biconnected graph must have *exactly* one source and one sink, as we now verify.

Lemma 1 *A 1-page book-embedding of a biconnected outerplanar graph has precisely one source and one sink.*

Proof Sketch. Assume that the embedding had more than one source (sinks yielding to a symmetric argument). The rightmost neighbors of all but the leftmost source are easily seen to be cut vertices, contradicting the biconnectivity of G. □-Lemma

□-Proposition

2. THE TYPENUMBERS OF SPECIFIC GRAPHS

We derive upper and lower bounds on the typenumbers of (book-embeddings of) a variety of families of graphs.

Stars. The d-ary *star* is the graph with $d+1$ vertices and with edges connecting one of these vertices (the *root*) to all the others (the *leaves*).

Proposition 5 *A connected graph G with at least two vertices has typenumber 2 iff G is a star.*

Proof Sketch. The one-page book-embedding of a star that places the root to one side (say the left) of all the leaves has two vertex types: the root has type $(0, d)$, and every leaf has type $(1, 0)$.

Conversely, let G be a connected graph with typenumber 2. By Proposition 3, G has at least one source and at least one sink. Since G has typenumber 2, it has precisely one source, say

$$\begin{pmatrix} 0 & R_1 \\ 0 & R_2 \\ & \vdots \\ 0 & R_p \end{pmatrix}$$

and precisely one sink, say

$$\begin{pmatrix} L_1 & 0 \\ L_2 & 0 \\ & \vdots \\ L_p & 0 \end{pmatrix}$$

If both $\sum_{i=1}^{p} L_i > 1$, and $\sum_{i=1}^{p} R_i > 1$, then G would not be a simple graph, as one can verify easily by considering the leftmost sink in the book-embedding and the source immediately to its left. If G is connected, and at least one of $\sum_{i=1}^{p} L_i$ and $\sum_{i=1}^{p} R_i$ equals 1, then G is a star. □

Complete Trees. For integers $d, h > 0$, the ***height-h complete d-ary tree*** $T_{d,h}$ is the rooted tree in which every nonleaf node has d sons, and all root-to-leaf paths have length h (measured in number of vertices). The ***preorder book-embedding*** of $T_{d,h}$ is the 1-page book-embedding whose vertex-ordering is given recursively by

0. the root of $T_{d,h}$, to the left of

1. the vertices of the leftmost copy of $T_{d,h-1}$ in preorder, to the left of

2. the vertices of the second leftmost copy of $T_{d,h-1}$ in preorder, to the left of

...

d. the vertices of the rightmost copy of $T_{d,h-1}$ in preorder.

Note that $T_{d,2}$ is the d-ary star.

Proposition 6 (a) *A 1-page book-embedding of a complete d-ary tree can have typenumber at most*

$$\begin{array}{rl} 1 & \text{if } h=1 \\ 3 & \text{if } h=2 \\ d+3 & \text{if } h=3 \\ d+5 & \text{if } h \geq 4. \end{array}$$

(b) *For any arity $d \geq 2$, the preorder book-embedding of $T_{d,h}$ (which is a 1-page book-embedding) has typenumber 1 when $h=1$, typenumber 2 when $h=2$, and typenumber 3 when $h \geq 3$. No book-embedding of $T_{d,h}$ has smaller typenumber.*

Proof. (a) Immediate from Proposition 4, since the root of $T_{d,h}$ is the unique node of degree d, all $(d^{h-1}-d)/(d-1)$ interior nodes have degree $d+1$, and all d^h-1 leaf nodes have degree 1.

(b) In the preorder book-embedding: the root has type $(0,d)$; each internal node has type $(1,d)$; each leaf node has type $(1,0)$. The typenumber-minimality of the preorder book-embedding follows from the lower bound in Proposition 4. □

Complete Graphs. The *n-vertex complete graph* K_n has n vertices, every two of which are connected by an edge.

Proposition 7 *Typenumber*$(K_n) = n$.

Proof Sketch. In any book-embedding of K_n, the i^{th} vertex from the left is unique in having precisely $i-1$ edges going to the left; hence, all vertices have distinct types. □

Complete Bipartite Graphs. For integers $m,n \geq 1$, the *complete bipartite graph* $K_{m,n}$ has m *input* vertices and n *output* vertices, and edges connecting each input with each output.

Proposition 8 *Typenumber*$(K_{m,n}) = 1+\min(m,n)$. *The book-embedding achieving this typenumber is unique, up to rotation.*

Proof. We show first that *Typenumber*($K_{m,n}$) $\leq 1 + \min(m,n)$. Say, with no loss of generality, that $m \leq n$. Consider the m-page book-embedding of $K_{m,n}$ that places all inputs to the left of all outputs and that uses page i $(1 \leq i \leq m)$ for the star that connects the i^{th} input vertex to all of the output vertices. In this embedding, each input vertex has a distinct type, and all output vertices have the same type, for a total of $1 + \min(m,n)$ types.

We complete the proof by showing that *Typenumber*($K_{m,n}$) $\geq 1 + \min(m,n)$, and that the book-embedding achieving this bound (which is the one described in the preceding paragraph) is unique, up to rotation.

Note first that the bound and the uniqueness are trivial when $\min(m,n) = 1$, for then $K_{m,n}$ is a star. We, therefore, assume henceforth that $\min(m,n) > 1$.

Focus on a fixed book-embedding of $K_{m,n}$. Say that there are two input vertices, u and v, that have the same type in the embedding. (Our conclusions will translate by symmetry to output vertices.)

Claim 1. No output lies between u and v in the book-embedding.

If u and v were separated by an output, they would have different numbers of leftgoing edges, hence different types.

Claim 2. For each output w, the edges from w to both u and v lie on the same page.

Claim 3. For each pair of outputs w and x, the edges from u to w and x lie on different pages; the same is true of the edges from v to w and x.

We verify Claims 2 and 3 simultaneously. By Claim 1, no output lies between inputs u and v in the book-embedding. Focus, therefore, on the k outputs that lie to the left of both u and v and on the l outputs that lie to the right of u and v. Say that the k lefthand outputs are, from left to right,

$$o_k, o_{k-1}, \cdots, o_1$$

and that the l righthand outputs are, from left to right,

$$\hat{o}_1, \hat{o}_2, \cdots, \hat{o}_l.$$

Say that, for each $i \in \{1, 2, \cdots, k\}$, input u's edge to output o_i resides on page p_i, and input v's edge to output o_i resides on page q_i; say, moreover, that, for each $j \in \{1, 2, \cdots, l\}$, input u's edge to output $\hat{o}_j$ resides on page $\hat{p}_j$, and input v's edge to output $\hat{o}_j$ resides on page $\hat{q}_j$. Suppose, with no loss of generality, that u lies to the left of v in the book-embedding. We note the following facts about the portion of the embedding that we have just described.

1. Since u and v share the same vertex type, we have both

   $$\{p_1, p_2, \cdots, p_k\} = \{q_1, q_2, \cdots, q_k\}$$

   and

   $$\{\hat{p}_1, \hat{p}_2, \cdots, \hat{p}_l\} = \{\hat{q}_1, \hat{q}_2, \cdots, \hat{q}_l\}.$$

2. Because edges on the same page cannot cross, we have:

   - for each $i \in \{1, 2, \cdots, k-1\}$, $q_i \notin \{p_{i+1}, \cdots, p_k\}$
   - for each $i \in \{1, 2, \cdots, l-1\}$, $\hat{p}_i \notin \{\hat{q}_{i+1}, \cdots, \hat{q}_k\}$
   - for each $i \in \{1, 2, \cdots, k\}$ and $j \in \{1, 2, \cdots, k\}$, $q_i \neq \hat{p}_j$.

   An easy inductive argument using these facts establishes that

   - for each $i \in \{1, 2, \cdots, k\}$, $p_i = q_i$
   - for each $i \in \{1, 2, \cdots, l\}$, $\hat{p}_i = \hat{q}_i$
   - for distinct $i, j \in \{1, 2, \cdots, k\}$, $p_i \neq p_j$
   - for distinct $i, j \in \{1, 2, \cdots, l\}$, $\hat{p}_i \neq \hat{p}_j$
   - for $i \in \{1, 2, \cdots, k\}$ and $j \in \{1, 2, \cdots, l\}$, $p_i \neq \hat{p}_j$

This Establishes Claims 2 and 3.

We now use Claims 1-3 to complete the proof.

Claim 4. If some two inputs have the same type in the book-embedding, then no two outputs have the same type.

By Claim 1, the outputs would have to lie on the same side of the similar-typed inputs; by Claims 2 and 3, the outputs would then have distinct types.

Claim 5. If some two inputs have the same type in the book-embedding, then the (common) type of these inputs differs from the types of all of the outputs.

Say that inputs u and v have the same type in the book-embedding. Focus on an arbitrary output w. By Claim 2, the edges from w to both u and v reside on the same page. By Claim 3, all edges emanating from u (and from v) reside on distinct pages. Since $\min(m, n) \geq 2$, this establishes the Claim.

In summary, we have shown that

- either all of the input vertices of $K_{m,n}$, or all of its output vertices, must have distinct vertex types in the book-embedding;
- if some two input (resp., output) vertices of $K_{m,n}$ have the same type in the embedding, then their common type must differ from the types of all of the output (resp., input) vertices.

It follows that there must be at least $1 + \min(m, n)$ distinct vertex types in the book-embedding. Moreover,

- the only way to achieve this minimum typenumber is
  - to place the more numerous of the inputs or outputs in a contiguous block
  - to dedicate a distinct page to (all of the edges incident to) each of the less numerous of the inputs or outputs.

This last item determines the type-minimizing book-embedding completely, up to rotation. (When $m = n$, we should also add the phrase "up to isomorphism".) □

Ladders. For integer $h \geq 1$, the ***height-h ladder graph*** L_h has vertex-set

$$\{a_1, a_2, \cdots, a_h, b_1, b_2, \cdots, b_h\}$$

and edges connecting

- each pair (a_i, b_i) for $i \in \{1, 2, \cdots, h\}$
- each pair (a_i, a_{i+1}) and each pair (b_i, b_{i+1}) for $i \in \{1, 2, \cdots, h-1\}$.

Proposition 9 *Restricting attention to 1-page book-embeddings of L_h:*

Typenumber$(L_1) = 2$;

Typenumber$(L_2) = 3$;

Typenumber$(L_3) = 4$;

for $h \geq 4$, *Typenumber*$(L_h) = 5$.

Proof. The result is trivial for $h < 3$: L_1 is the 1-ary star, so the result is a special case of Proposition 5; L_2 is the 4-cycle, so its unique 1-page book-embedding has 3 vertex types, which must be minimal since L_2 is not a star.

Focus henceforth on the case $h \geq 3$.

Consider the 1-page book-embedding of L_h that orders the vertices

$$a_2 - a_3 - \cdots - a_h - b_h - \cdots - b_2 - b_1 - a_1.$$

When $h = 3$, this embedding uses the four vertex types

$$\{(0,3), (1,1), (2,0), (2,1)\}.$$

When $h = 4$, this embedding uses the five vertex types

$$\{(0,3), (1,1), (1,2), (2,0), (2,1)\};$$

the rotations of this book-embedding also use five vertex types (but, different ones). This establishes the upper bound.

To see the lower bound, note first that L_h is a biconnected outerplanar graph; hence by Lemma 1, a 1-page embedding of L_h has precisely one source and one sink. Since L_h has four bivalent vertices, at least two of these must have type (1,1).

Consider first the case $h = 3$. We have accounted for three types thus far, the type (1,1), the source and the sink. We consider three cases.

1. If the source and sink are both bivalent vertices, then we need at least one more vertex type, to account for the two trivalent vertices in L_3.
2. If the source and the sink are both trivalent, then we cannot have a legal book-embedding, for one cannot have on a single page two vertex-disjoint length-4 paths with the same endpoints.
3. If one of the source and sink is bivalent, while the other is trivalent, then we need a fourth vertex type, to account for the second trivalent vertex.

In all realizable cases, then, we need four vertex types.

When $h \geq 4$, we need one more ingredient for our argument. Let $n(L, R)$ denote the number of vertices in the embedding that have type (L, R). Since across the entire embedding, the number of edges "leaving some vertex to the right" must equal the number of edges "leaving some vertex to the left", we must have the equation

$$2n(0,2) + 3n(0,3) + n(1,2) = 2n(2,0) + 3n(3,0) + n(2,1).$$

Since we now have at least four trivalent vertices, this conservation equation must be satisfied subject to the following equalities:

- $n(0,2) + n(1,1) + n(2,0) = 4$
- $n(0,3) + n(1,2) + n(2,1) + n(3,0) = 2h - 4$
- $n(0,2) + n(0,3) = n(2,0) + n(3,0) = 1$

One shows easily that this system of equalities can be satisfied only if

$$n(1,2) \cdot n(2,1) > 0.$$

This means, however, that we have at least five vertex types in the book-embedding: the three two-sided types plus the source and the sink. The lower bound follows. □

We conjecture that additional pages for L_h, $h \geq 4$, will not lower its typenumber below 5, but we suspect that only an unilluminating case analysis will settle the conjecture.

3. TYPENUMBER AND GRAPH STRUCTURE

We have tried to link the typenumber of a graph with some other structural property of the graph. With the exception of a nontrivial bound connecting the number of pages and the number of types in a book-embedding, our quest has been unsuccessful.

3.1. Book-Thickness

We begin with our one success. Since trees, which are 1-page embeddable, can have arbitrarily large sets of distinct vertex-degrees, hence arbitrarily large typenumbers, one cannot hope to bound the book-thickness of G from below using the typenumber of G. But, one can obtain a nontrivial upper bound on the book-thickness of G from the typenumber.

Proposition 10 *If the graph G admits a p-page book-embedding with t vertex types, then*

$$p \leq \binom{t}{2};$$

no smaller bound is generally possible, since this bound is achievable.

Proof. Consider an arbitrary p-page t-type book-embedding. Say that page k $(1 \leq k \leq p)$ is *introduced* by the leftmost vertex v in the embedding that has an edge entering it from the left on page k, i.e., that has a positive entry L_k in its vertex type $\tau(v)$ as in (1). Assume that vertex v of G introduces both page i and page j in a book-embedding. Then the vertices u_1 and u_2 that lie to the left of v in the embedding and that "emit" the witnessing edges on pages i and j must have distinct vertex types: If they had the same type, then the fact that edges on a given page of a book-embedding do not cross would force v to accept both of these edges from the rightmore of u_1 and u_2; these "parallel edges" would contradict the simplicity of G. It follows that a vertex in the embedding can introduce no more pages than it has distinct vertex types to its left in the embedding. Furthermore, if v introduces a page, then its type differs from the types of all the vertices to its left. It follows that the m^{th} vertex from the left in the book-embedding can introduce no more than $m-1$ pages (and this many only if all vertices to its left have distinct types). The bound follows.

To see that the bound is achievable, consider K_n. We saw in Proposition 7 that every book-embedding of K_n has n vertex types. Consider the (thickness-inefficient) book-embedding that assigns each edge of K_n to a distinct page. This embedding uses n types and $\binom{n}{2}$ pages. □

Despite whatever hopes our bound might raise, one finds that one cannot generally hope to move toward optimizing either typenumber or book-thickness by moving toward optimizing the other.

Proposition 11 (a) *There exist graphs G and t-type p-page book-embeddings of G such that every $(p+1)$-page book-embedding of G has more than t types.*
(b) *There exist graphs G such that every p-page book-embedding of G uses at least $t+1$ types, but there is a $(p+1)$-page book-embedding of G that uses fewer than t types.*

Proof Sketch. (a) There is a 1-page 2-type book-embedding of the d-ary star (Proposition 5). For $d > 1$, any $(p > 1)$-page book-embedding must have more than 2 types, since some leaves must reside on distinct pages, hence have distinct types.

(b) By Proposition 8, the book-embedding that minimizes the typenumber of $K_{n,n}$ is unique, up to rotation. This $(n+1)$-type embedding uses $n+1$ pages. Bernhart and Kainen [1] exhibit an n-page book-embedding of $K_{n,n}$. It follows that type-minimality cannot be achieved simultaneously with thickness-minimality. □

3.2. Graph Homeomorphism

In the case of graph homeomorphism, not only can we show that taking homeomorphs can either increase or decrease typenumber, we can show that no functional bound can be placed on the amount of change.

Proposition 12 (a) *There exist graphs G and t-type book-embeddings of G such that every nontrivial homeomorph of G admits only book-embeddings that use more than t types.*
(b) *Let D_G denote the set of distinct vertex-degrees in the graph G. There exists a homeomorph of G that admits a 3-page book-embedding with $|D_G|+14$ vertex types. Thus, there exist graphs G such that every book-embedding of G uses at least t types, but there is a homeomorph of G one of whose book-embeddings uses strictly fewer than t types.*

Proof Sketch. (a) We invoke the d-ary star once more. Any nontrivial homeomorphism of the star is no longer a star, hence cannot be realized with a 2-type book-embedding.

(b) We modify a device of Bernhart and Kainen [1, Theorem 5.4]. Let $G = (V, E)$ be a simple undirected graph. We construct a 3-page-embeddable homeomorph of G in stages, by constructing the book-embedding directly:

1. Lay the $|V|$ vertices of G along the spine of the book, in arbitrary order.
2. Place $|E|$ new vertices along the spine, to the right of G's vertices.
3. Use one page to lay the edges of a forest of stars, with G's vertices as roots and the new vertices as leaves: each d-valent vertex of G becomes the root of a d-ary star. In order to fit on one page of the book, these stars must be nested in the sense that G's rightmost vertex uses the leftmost new vertices, G's second rightmost vertex uses the new vertices just to the right of these leftmost ones, and so on.

4. Using one "pseudo-page" – a page that allows crossed edges – connect up the new vertices so as to construct a homeomorph of G, in fact one that has precisely two new vertices along each edge of G.

5. Place a pseudo-vertex at each edge-crossing on the pseudo-page. Distort the edges on the pseudo-page so that all of the pseudo-vertices lie along a line. Place a new pseudo-vertex at each point where an edge crosses the line.

6. Pull the line of pseudo-vertices around 180 degrees rigidly, so that the pseudo-vertices lie on the spine, to the right of the new vertices.

7. Add new pseudo-vertices along the spine to the right of the already-present pseudo-vertices, wherever an edge crosses the spine.

One now has a 2-page book-embedding of a graph derived from G. At this point, one can apply the techniques of Theorem 5.4 of [1] to use the third page to modify this derived graph so that it becomes a homeomorph of G. (We refer the reader to that paper rather than repeat the construction.)

The 3-page book-embedded homeomorph has the following characteristics.

- The leftmost $|V|$ vertices have $|D_G|$ different valences; every degree-d vertex in the block has type

$$\begin{pmatrix} 0 & d \\ 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

- All other vertices in the homeomorph are bivalent (as they must be); their types are arbitrary, except that they cannot contain any entry "2", nor can they contain the pattern "(1,1)" on page 3.

The result follows by adding up the number of types. □

4. ON MINIMIZING TYPENUMBER

4.1. The Infeasibility of Minimizing Graph-Typenumber

We now show that the problem of finding a typenumber-minimal book-embedding of a given graph is likely to be computationally infeasible. We accomplish this by considering the following decision problem.

MIN-OUTERPLANAR-TYPES:

Instance: An outerplanar graph G and an integer k.

Question: Does G have a 1-page embedding using at most k vertex types?

Theorem 1 *The problem* **MIN-OUTERPLANAR-TYPES** *is* NP*-complete.*

Proof. It is transparent that **MIN-OUTERPLANAR-TYPES** is in *NP*, so we shall concentrate only on showing that it is *NP*-hard. To this end, we shall begin by showing that the following problem, **MIN-UNION**, is polynomial-time reducible to **MIN-OUTERPLANAR-TYPES**. We shall then complete the proof by showing that **MIN-UNION** is *NP*-complete.

MIN-UNION:

Instance: m pairs (A_i^1, A_i^2) of subsets of $\{1, 2, \cdots, n\}$ and an integer k.

Question: Are there choices $j_i \in \{1, 2\}$, $1 \le i \le n$, such that $|\bigcup_i A_i^{j_i}| \le k$?

In the following, we specify outerplanar graphs, and portions of them, by presenting a sequence of vertex types; Proposition 2 justifies this mode of specification. The notation does not, of course, imply that the given order is the one used in embedding the graph.

Let an instance of **MIN-UNION** be given. Let $s(i)$ be the vertex sequence[2] $(1,2)^{i+2}(1,1)(i+3,1)$, and let $\hat{s}(i)$ be the reverse sequence $(1,i+3)(1,1)(2,1)^{i+2}$. To each pair

$$(A_i^1 = \{j_1^1, j_2^1, \cdots, j_{k_i^1}^1\}, A_i^2 = \{j_1^2, j_2^2, \cdots, j_{k_i^2}^2\})$$

associate two copies of the biconnected component

$$B_i = (0,2),(1,1),s(j_1^1+3),s(j_2^1+3),\cdots,s(j_{k_i^1}^1+3),\hat{s}(j_1^2+3),\hat{s}(j_2^2+3),\cdots,\hat{s}(j_{k_i^2}^2+3),(2,0). \tag{4}$$

In addition take two copies of

$$E_1 = (0,2),(1,1),\hat{s}(4),\hat{s}(5),\cdots,\hat{s}(n+3),(2,0). \tag{5}$$

If $n > 2m-2$, take $\lceil (n-2m+2)/2 \rceil$ copies of (0,2),(1,1),(2,0). Join all of the above components into one component by identifying the first vertex of every sequence; call the resulting vertex, with degree at least $n+4$, the *anchor* vertex. Finally, add the two connected components

$$E_2 = (0,2),(1,1),(2,0) \tag{6}$$

and

$$E_3 = (0,2),(1,2),(1,1),(2,1),(1,2),(1,1),(2,1),(2,0). \tag{7}$$

The preceding construction results in an outerplanar graph G that has a one-page embedding using $k+n+6$ types if and only if there is a choice of sets from each pair (A_i^1, A_i^2) with a union of size k, as we show now.

Suppose that there is a vector of choices of sets, $\bar{l} = (l_1, l_2, \cdots, l_m) \in \{1,2\}^m$, satisfying $|\bigcup_i A_i^{l_i}| \le k$. Then G can be embedded using $k+n+6$ types, as follows. The components

[2] $(a,b)^c$ denotes a string of c occurrences of the pair (a,b).

E_1, E_2, and E_3 are embedded as specified in (5,6,7) above. If $l_i = 1$, then both copies of B_i are embedded in the order specified in (4) above. If $l_i = 2$, then both copies of B_i are embedded in the *reverse* of the order specified in (4). One verifies easily that the embedding always uses types

$$\{(1,4),(1,5),\cdots,(1,n+3),(2,0),(1,1),(0,2),(1,2),(2,1)\}$$

for E_1, E_2, and E_3, and some type (a_1, a_2) where $a_1 + a_2 \geq n + 4$ for the anchor vertex. Let $\bigcup_i A_i^{l_i} = \{r_1, r_2, \cdots, r_k\}$. Then the only additional types from the B_i are $\{(r_i + 3, 1) : 1 \leq i \leq k\}$. Hence the total number of types is $k + n + 6$. (Note that the embedding of the component E_1 "swallows up" the types corresponding to the elements A_j^i that are not selected by the vector $\bar{l}$.)

Next, suppose that G can be embedded in one page using at most $k + n + 6$ types. Because each component of G is biconnected, the possible one-page embeddings G are very restricted, as the following Lemma indicates.

Lemma 2 [14] *A biconnected outerplanar graph has a unique outerplanar embedding, up to rotation and reversal.*

For any embedding of G, the components E_2 and E_3 must use types (0,2), (2,0), (1,1), (1,2), and (2,1). In addition, the anchor vertex is of some type (a_1, a_2) where $a_1 + a_2 \geq n+4$. Of the two copies of E_1, at least one must have the anchor vertex at one end. Hence, we may assume without loss of generality that types $\{(1, i+3) : 1 \leq i \leq n\}$ are used also.

For $1 \leq i \leq n$, one copy of B_i must have the anchor vertex at one end. That copy must use either types $\{(j_i^1, 1) : 1 \leq i \leq k_i^1\}$ or $\{(j_i^2, 1) : 1 \leq i \leq k_i^2\}$. In either case, $|\bigcup_i A_i^{l_i}| \leq k$ for the corresponding selection vector $\bar{l}$.

We have thus reduced the **MIN-UNION** problem to the **MIN-OUTERPLANAR-TYPES** problem. We now complete our proof by establishing the *NP*-completeness of **MIN-UNION**. As before, membership in *NP* is transparent, so we shall show only that the problem is *NP*-hard, by reducing the well-known *NP*-complete problem **CLIQUE** [6] to it.

CLIQUE:

Instance: An undirected graph $G = (V, E)$ and an integer k.

Question: Does G have a clique of size at least k?

Assume, without loss of generality, that $V = \{1, 2, \cdots, n\}$. For $1 \leq i \leq n$, let $A_i^1 = \{i\}$ and $A_i^2 = \{j : (j, i) \notin E\}$. Given a selection vector $\bar{l} = (l_1, l_2, \cdots, l_m) \in \{1,2\}^m$, define $V_1^{\bar{l}} = \{i \in V : l_i = 1\}$, $V_2^{\bar{l}} = \{i \in V : l_i = 2\}$, and $A^{\bar{l}} = \bigcup_i A_i^{l_i}$.

We claim that G has a clique of size k if, and only if, there is a selection vector $\bar{l}$ such that $|A^{\bar{l}}| \leq n - k$.

Suppose first that the graph G has a clique $C \subseteq V$. Let $l_i = 2$ for $i \in C$ and $l_i = 1$ for $i \notin C$. Then $A^{\bar{l}} \subseteq V - C$; hence, if C has size k, then $|A^{\bar{l}}| \leq n - k$.

Conversely, if $|A^{\bar{l}}| \leq n - k$, then the set $V_2^{\bar{l}}$ contains a clique of size k. This can be seen as follows. Each $i \in V_1^{\bar{l}}$ appears in $A^{\bar{l}}$, hence adds 1 to $|A^{\bar{l}}|$. A given $j \in V_2^{\bar{l}}$ fails to appear in $A^{\bar{l}}$ if, and only if, it is adjacent to every other vertex of $V_2^{\bar{l}}$. In particular, some k vertices from $V_2^{\bar{l}}$ fail to appear in $A^{\bar{l}}$ if, and only if, they are all mutually adjacent, hence form a clique.

The argument in the preceding paragraphs establishes that **CLIQUE** is reducible to **MIN-UNION**, so **MIN-UNION** is *NP*-complete. This completes the proof of the Theorem. □

As an added point of interest, the **MIN-UNION** problem is closely related to a variation of the **SATISFIABILITY** problem: Given an instance (A_i^1, A_i^2) of **MIN-UNION**, construct a CNF formula with clauses

$$C_j = \{x_i \ : \ i \in A_j^1\} \cup \{\neg x_i \ : \ i \in A_j^2\}.$$

There is a selection vector $\bar{l}$ for which the union set $A^{\bar{l}}$ has cardinality at most k if, and only if, there is an assignment of truth values to the x_i that satisfies at most k clauses. This **MIN-CNF-SAT** problem contrasts with the **MAX-2CNF-SAT** problem, which is proved *NP*-complete in [7].

4.2. The Importance of Minimizing Embedding-Typenumber

Of the three measures of the cost of a book-embedding, typenumber is the only one that is very sensitive to rotation (or, cyclic permutation). It is obvious that rotation has no effect on the book-thickness of a book-embedding. It is not difficult to verify that rotation can increase (or decrease) the pagewidth of a book-embedding by at most a factor of 2; this observation is due to L. S. Heath (and is used in [4]). However, there is no *a priori* bound on how much the typenumber of a book-embedding can be affected by rotation.

To verify this last claim, focus on the (1-page) preorder book-embedding of the height-h complete d-ary tree $T_{d,h}$. We showed in Proposition 6(b) that this embedding never uses more than 3 vertex types. Moderating this good news (and emphasizing the intended message of this subsection) is the fact that no nontrivial rotation of the preorder book-embedding has typenumber less than 4: any rotation uses at least 3 types for the nonroot nodes of the tree. Even worse, a positive fraction of the possible rotations of the preorder book-embedding actually use the pessimal number $d + 5$ of types. We now verify this latter assertion.

Consider the familiar labelling of the nodes of $T_{d,h}$ with strings: Let the root node of $T_{d,h}$ be labelled by the null string, and inductively let the d children of the node labelled by the string ν be labelled by the strings $\nu 1, \nu 2, \cdots, \nu d$, in left-to-right order of their appearance in the preorder book-embedding. It will be useful to determine how many of these string labels contain all of the "letters" in the set $\{1,2,\cdots,d-1\}$.

Lemma 3 *Let $A=\{1,2,\cdots,d\}$. The number of length-n strings of the form αx, where $\alpha \in A$ and where x is a length-$(n-1)$ string that contains all of the letters $\{1,2,\cdots,d-1\}$ in some order, is precisely*

$$S(n;d)=\sum_{i=0}^{d-1}(-1)^i\binom{d-1}{i}\cdot d\cdot(d-i)^{n-1}$$

Proof. We invoke the Principle of Inclusion and Exclusion [11, Ch. 3]: There are d^n length-n strings over the alphabet A; there are $d-1$ properties $P_j(x)\equiv[j$ does not appear in $x]$; for each $1\le i\le d-1$,

$$\binom{d-1}{i}\cdot d\cdot(d-i)^{n-1}$$

of these length-n strings have the form αx, where $\alpha\in A$ and where x lacks at least i of the letters $\{1,2,\cdots,d-1\}$. □

We now verify that certain leaves whose labels are close to the form enumerated in Lemma 2 lead to typenumber-pessimal rotations of the preorder book-embedding of $T_{d,h}$, while arbitrary nodes whose labels have that form lead to rotations that are close to pessimal.

Proposition 13 *Consider the preorder book-embedding of $T_{d,h}$, where $h>d\ge 2$. If we rotate the embedding so that the leftmost node, called the* pivot *node, is one of the $S(h-2;d)$ leaves whose label has the form $\alpha x1$, where $\alpha\in\{1,2,\cdots,d\}$ and where x is a length-$(h-3)$ string that contains all of the letters $\{1,2,\cdots,d-1\}$ in some order, then the resulting book-embedding has $d+5$ vertex types. For fixed d, as h grows without bound, these $S(h-2;d)$ leaves approach the fraction $(d-1)/d^2$ of the nodes of $T_{d,h}$.*

Proof Sketch. Consider the preorder book-embedding, rotated as prescribed in the statement of the Proposition. Note that when the pivot node ν has a label of the prescribed form, then in the preorder book-embedding it has, for each $i\in\{0,1,2,\cdots,d-1\}$, a node to its left that has precisely i children to ν's right. The edges to these children change from rightgoing to leftgoing after the prescribed rotation. Using this fact one can verify that in the prescribed rotation of the preorder embedding: the root of $T_{d,h}$ has type $(k,d-k)$ for some $0\le k\le d$; the pivot leaf has type (0, 1), while all leaves that were to its left before the rotation have type (1, 0); all internal nodes that were to the left of the pivot leaf and whose children were also to the left of the pivot leaf before the rotation have type $(1,d)$; the father of the pivot leaf has type $(d+1,0)$; each internal node that lay to the left of the pivot leaf before the rotation but had i ($i\in\{1,2,\cdots,d-1\}$) children to the right of the pivot leaf has type $(i+1, d-i)$, while its i children have type $(0, d+1)$. Summarizing, we find the following types:

$$\begin{array}{cl} (k, d-k) & \text{the root} \\ (i, d-i+1) & 0 \leq i \leq d+1 \text{ the internal nodes} \\ (0,1) & \text{the pivot leaf} \\ (1,0) & \text{all other leaves} \end{array} \tag{8}$$

which are $d+5$ in number.

Finally, note that the claimed cardinality of the set of bad pivot leaves follows immediately from Lemma 2. □

In fact, the situation is even worse than Proposition 13 suggests: Asymptotically as we consider successively taller complete d-ary trees, rotating the preorder book-embedding to a random site will, with probability approaching 1, yield an embedding with at least $d+4$ vertex types. This follows from the fact that the overwhelming majority of the nodes of $T_{d,h}$ have labels of the form prescribed in Lemma 2, and each is a bad pivot node.

Proposition 14 *Consider the preorder book-embedding of $T_{d,h}$, where $h > d \geq 2$. If we rotate the embedding so that the pivot node has a label of the form αx, where $\alpha \in \{1, 2, \cdots, d\}$ and where x is a string that contains all of the letters $\{1, 2, \cdots, d-1\}$ in some order, then the resulting book-embedding has at least $d+4$ vertex types. For fixed d, as h grows without bound, these "bad" pivot nodes constitute the fraction*

$$1 - O\left(\left(\frac{d-1}{d}\right)\right)$$

of the nodes of $T_{d,h}$.

Proof Sketch. Each length-n string referred to in Lemma 2 is the label of some node at level n of $T_{d,h}$. By the reasoning in the proof of Proposition 13, the typenumber of the preorder book-embedding of $T_{d,h}$, rotated so that any such node is the pivot node, is at least $d+4$: all of the types enumerated in (8), save perhaps (0, 1), must occur. We can, therefore, use Lemma 2 to calculate a lower bound on the number of nodes of $T_{d,h}$ whose labels guarantee such large typenumber. By our earlier reasoning, this number is no less than

$$\begin{aligned}
\sum_{n=d+1}^{h-1} S(n;d) &= \sum_{n=d+1}^{h-1} \sum_{i=0}^{d-1} (-1)^i \binom{d-1}{i} \cdot d \cdot (d-i)^{n-1} \\
&= d \cdot \sum_{i=0}^{d-1} (-1)^i \binom{d-1}{i} \sum_{n=d+1}^{h-1} (d-i)^{n-1} \\
&= d \cdot \sum_{i=0}^{d-2} (-1)^i \binom{d-1}{i} \cdot \left(\frac{(d-i)^{h-1} - (d-i)^d}{d-i-1}\right) + (-1)^{d-1} \cdot d \cdot (h-d-1).
\end{aligned}$$

Since $T_{d,h}$ has $(d^h - 1)/(d-1)$ nodes, the claimed fraction of the pivot nodes follows. □

In order to give the reader a feeling for how fast the "eventually" of the big-O in Proposition 14 eventuates, we present a short table that indicates, as a function of d and h, the fraction of potential pivot nodes in $T_{d,h}$ whose labels satisfy the conditions of the Proposition; all lead to book-embeddings with typenumber at least $d+4$.

Propositions 13 and 14, being asymptotic, are at most suggestive in impact. We now present two tables that indicate the maximum, minimum, and average typenumber observed in a series of experiments that looked at all rotations of a variety of book-embeddings. We rotated the preorder book-embedding of $T_{d,h}$, for a variety of values of d and h, and we rotated the (thickness- and width-optimal) 2-page book-embedding from [3] of the height-h X-tree $X(h)$, for a few values of h. The *height-h X-tree* is obtained from the height-h complete binary tree $T_{2,h}$ by adding edges linking all nodes at each level in a line. For perspective, we present an analog of Proposition 6 for X-trees.

Proposition 15 *A 2-page book-embedding of a height-h X-tree can have typenumber at most*

$$\min(56, 2^{h-1} - 2h + 2) + \min(35, 2h - 4) + \min(20, 2^{h-1} - 2) + 3.$$

Proof Sketch. Immediate from Proposition 4, given that the height-h X-tree has $2^{h-1} - 2h + 2$ nodes of degree 5, $2h - 4$ nodes of degree 4, $2^{h-1} - 2$ nodes of degree 3, and 3 nodes of degree 2. □

Tables 2 and 3 demonstrate that the phenomenon that is perhaps exaggerated by the asymptotics of Propositions 13 and 14 obtains even for very modest size situations, at least for the indicated book-embeddings of both complete trees and X-trees.

The results of this subsection establish the need for the algorithm of the next section.

4.3. Rotating a Book-Embedding to Minimize Typenumber

We now describe a family of efficient algorithms that find that rotation of a book-embedding that minimizes typenumber. All of the algorithms follow the same strategy; they differ in their data structures, which are tailored to the magnitudes of the book-thickness p of the book-embedding and the maximum vertex-degree d of the embedded graph G.

The Algorithmic Strategy

Input. A p-page book-embedding of an n-vertex graph G, presented via the associated sequence $\tau_1, \tau_2, \cdots, \tau_n$ of vertex types.

Output. A list, with one entry for each vertex v of G. The entry for each v comprises

- the number of vertex types in the book-embedding obtained by rotating the input embedding so that v becomes the pivot vertex;

- The *bag* (multiset, with multiplicities) of vertex types that appear in the book-embedding when v is the pivot vertex.

The Strategy. Say that vertex v is the current pivot vertex of the book-embedding. In order to rotate the embedding, we do the following.

Step 1. Rotate v's $p \times 2$ type-matrix so that each row $(0, R_k)$ becomes $(R_k, 0)$; this effectively moves v to the right end of the embedding.

Step 2. For each vertex w that is adjacent to v, if the edge of adjacency resides on page k, then row k of w's type-matrix is changed from (L_k, R_k) to $(L_k - 1, R_k + 1)$; this effectively "swings that edge around".

Step 3. As each vertex type is altered in Steps 1 or 2, the old type's multiplicity is *diminished* in the bag and the new type's multiplicity is *augmented* in the bag; when a type disappears from the bag, or appears for the first time (detected by the multiplicities), the typecount of v is adjusted accordingly.

Implementing the Strategy

We need to specify the data structures used to implement the strategy.

The Paged Adjacency Table.

It is convenient to make use of a *paged adjacency table* for G, i.e., an adjacency list with a record of which edges lie on which pages. Such a list is readily constructed from the input. Two organizations for the list recommend themselves, the choice being dictated by the magnitudes of p (the book-thickness of the input embedding) and of the number of edges of the input graph G. In both representations, there is a one-dimensional array (the table) whose entries are the vertices of G, in the order they lie along the spine of the book. The table entry corresponding to vertex v of G comprises

- a register $Type\_Count(v)$ recording the number of vertex types in the book-embedding when v is the pivot vertex;
- a pointer into the bag of vertex types that points, at any given moment, to the then-current vertex type of v;
- a pointer to a list of those vertices that are adjacent to v in G; each such vertex is represented via a pointer into the table.

The two organizations differ in their representations of the information about which edges lie on which pages.

Distributed Page Information. In this organization, each vertex w in the list of vertices adjacent to vertex v contains a field indicating the page via which v is adjacent to w. Since each such field requires $\log_2 p$ bits, and since there is such a field for each endpoint of each of G's e edges, this organization requires

$$2e \cdot \log_2 p$$

bits for recording the page information.

Centralized Page Information. In this organization, the entry of each vertex v in the table contains p subfields; the i^{th} subfield contains a pointer to a list of those vertices that are adjacent to v in G via page i. This creates $(p-1)n$ new subfields in all, each of $\log_2 n$ bits, since it must be capable of pointing to any vertex of G. Thus, this organization requires

$$(p-1)n \cdot \log_2 n$$

bits for recording the page information.

For all but quite dense graphs, the distributed organization is likely to be the preferred one.

The Bag of Types.

We must represent the bag of vertex types in a way that facilitates adding, deleting, and altering types, where each alteration either exchanges the L- and R-entries of a type-matrix (thereby moving the associated vertex from the left end to the right end of the book-embedding), or adds $(-1,+1)$ to some row of the type-matrix (thereby flipping one edge from left-entering to right-entering). As noted earlier, the changes to the bag must maintain an accurate count of the multiplicity of each vertex type in the bag. The preferred representation of a bag depends on the magnitudes of the book-thickness p of the book-embedding and the maximum vertex-degree d of G.

A. *p and d both small.*

This case is quite common:

binary trees	$p=1$	$d=3$	[4]
2 − dimensional meshes	$p=3$	$d=4$	[4]
X − trees	$p=2$	$d=5$	[4]
Benes networks	$p=3$	$d=4$	[5]

In this case, equation (3) assures us that the number of vertex types must be small, in fact cannot exceed

$$\binom{2p+d}{d} - 1;$$

we shall, therefore, represent the bag as a one-dimensional *array* of nonnegative integers, indexed as follows. The array/bag-entry corresponding to vertex type

$$\begin{pmatrix} L_1 & R_1 \\ L_2 & R_2 \\ & \vdots & \\ L_p & R_p \end{pmatrix},$$

each $L_i, R_i \in \{0, 1, \cdots, d\}$, is determined by converting the type-matrix to the length-$2p$ $(d+1)$-ary numeral

$$L_p R_p L_{p-1} R_{p-1} \cdots L_1 R_1,$$

and evaluating the numeral; since p is fixed, distinct numerals specify distinct numbers. Each entry in the array is the multiplicity of the vertex type in the book-embedding that indexes that entry.

At any given moment, we shall remember the index of the current pivot vertex v of the book-embedding. In order to rotate the book-embedding one place, we move the current v to the right end of the embedding, and we use the paged adjacency table to pick up the type of the next pivot vertex (which is the successor in the table of the current pivot vertex). Moving v to the right end of the embedding consists of the following.

Step 1. Decrease the multiplicity of v's current vertex type.

Step 2. Determine v's new vertex type, which is obtained by multiplying the current type by $d + 1$ (recall that all current L_i are 0 and that we want to flip each L_i and R_i).

Step 3. Change the vertex type of each neighbor w of v, after decreasing the multiplicity of w's current type; if vertex w is adjacent to v on page k (discovered from the paged adjacency table) then the new vertex type of w is obtained by decreasing the current type by the quantity $d(d+1)^{2k-2}$ (recall that we want to decrease L_k by 1 and increase R_k by 1).

Of course, during these alterations we are keeping track of any vertex types that disappear or appear anew, to evaluate *Type_Count*(v).

B. *p large and d small.*

Although we do not know of any specific graphs with such p and d, we do know that they exist: It is proved in [4] that for all d and all sufficiently large n, there exist n-vertex graphs with maximum degree d that cannot be embedded in fewer than

$$(const)\frac{n^{1/2-1/d}}{log^2 n}$$

pages. Since p is large, the array-solution of the previous subsection is likely to be too wasteful of memory, since the bag/array is likely to be sparsely occupied. We shall, therefore, represent our bag of vertex types as a height-$(2p+1)$ *trie* (digital search tree).

Each internal node of the trie will be a $(d+1)$-ary array of pointers to the adjacent levels of the trie; we assume that all edges are bidirectional. The array positions represent labels on the edges from a node to its children, the labels $0, 1, \cdots, d$ being the valid entries in a vertex type. Each vertex type that occurs in the current rotation of the book-embedding will be a root-to-leaf path in the trie; the multiplicity of the type will be recorded in its leaf node.

At any given moment, we shall point to the leaf-node corresponding to the current vertex type of the current pivot vertex v of the book-embedding. In order to rotate the book-embedding one place, we move the current pivot v to the right end of the embedding, and we use the paged adjacency table to pick up the type of the next pivot vertex. Moving v to the right end of the embedding consists of the following.

Step 1. Decrease the multiplicity of v's current vertex type.

Step 2. Determine v's new vertex type, which is obtained by proceeding along the (unique) path from v's leaf to the root of the trie, constructing the vertex type "complementary" to v's, i.e., converting v's current type

$$\begin{pmatrix} 0 & T_1 \\ 0 & T_2 \\ & \vdots \\ 0 & T_p \end{pmatrix}$$

to v's new type

$$\begin{pmatrix} T_1 & 0 \\ T_2 & 0 \\ & \vdots \\ T_p & 0 \end{pmatrix}$$

if the multiplicity of v's current type was decreased to 0 in Step 1, then, as a space-saving measure, all trie-entries unique to this vertex type can be removed during the leaf-to-root traversal (i.e., all trie-nodes can be removed, up to the occurrence of the first binary node along the path).

Step 3. Record v's new vertex type by traversing the root-to-leaf path in the trie dictated by the new vertex type, inserting new nodes when necessary, and increasing this type's multiplicity when the leaf node is reached.

Step 4. Change the vertex type of each neighbor w of v, after decreasing the multiplicity of w's current type; this is accomplished by traversing the path from the leaf containing w's vertex type toward the root, until one encounters the type-entry for the page on which v is adjacent to w (as in Step 3, node entries for a zeroed type can be removed during this traversal); one now reverses direction, following the root-to-leaf path dictated by w's new type, adding new nodes when necessary, and increasing this new type's multiplicity when the leaf node is reached.

C. *p and d both large.*

When both p and d are large, as with the complete graphs K_n or complete bipartite graphs $K_{m,n}$, any data structure that incorporates arrays is likely to be wasteful of space.

The trie structure of the previous subsection deals well with large p; with one small modification, it accommodates large d also. Rather than have a $(d+1)$-entry array of pointers at each nonleaf node of the trie, we shall now have a balanced binary tree, having up to $\lceil \log_2(d+1) \rceil$ levels. The processing of the trie proceeds as in the previous subsection, with the one complication that, as vertex types are added or deleted, the nonleaf nodes' balanced trees are dynamically updated. Details are left to the reader.

Analyzing the Implementation

As the algorithm proceeds, the vertex type of a degree-d vertex v of G is changed $d+1$ times: once as v is moved from the left end to the right end of the book-embedding, and once as each of its neighbors is so moved. The structure of the paged adjacency table allows one to determine in time $O(1)$ what vertex type to access, how to access it, and what change to make. The cost of maintaining the bag of vertex types depends on the magnitudes of p and d; even with our trie of trees data structure, which is the most costly of the three we describe, each transaction involved in altering a type can be done in time proportional to $p \cdot \log d$. If the graph G has n vertices, e edges, and the set D of distinct vertex-degrees, then the entire algorithm can be executed within time proportional to

$$\sum_{d \in D} p \cdot (d+1) \cdot \log d$$

which is in turn proportional to

$$p \cdot e \cdot \log d_{max}.$$

ACKNOWLEDGMENT. This research was supported in part by SRC Contract 85-02-054; the research of the second author was supported in part also by NSF Grant DMC-85-04308. The authors wish to thank Reuven Bar Yehuda, Sandeep Bhatt, and Lenny Heath for helpful and stimulating conversations concerning the material in this paper.

5. REFERENCES

1. F. Bernhart and P.C. Kainen (1979): The book thickness of a graph. *J. Comb. Th. (B) 27,* 320-331.

2. J.F. Buss and P.W. Shor (1984): On the pagenumber of planar graphs. *16th ACM Symp. on Theory of Computing,* 98-100.

3. F.R.K. Chung, F.T. Leighton, A.L. Rosenberg (1983): DIOGENES – A methodology for designing fault-tolerant processor arrays. *13th Intl. Conf. on Fault-Tolerant Computing,* 26-32.

4. F.R.K. Chung, F.T. Leighton, A.L. Rosenberg (1987): Embedding graphs in books: A layout problem with applications to VLSI design. *SIAM J. Algebr. Discr. Meth. 8,* 33-58.

5. R.A. Games (1986): Optimal book embeddings of the FFT butterfly, Benes, and barrel shifter networks. *Algorithmica,* to appear.

6. M.R. Garey and D.S. Johnson (1979): *Computers and Intractability*, Freeman, New York.

7. M.R. Garey, D.S. Johnson, L.J. Stockmeyer (1976): Some simplified NP-complete graph problems. *Theoret. Comput. Sci. 1*, 237-267.

8. L.S. Heath (1984): Embedding planar graphs in seven pages. *25th IEEE Symp. on Foundations of Computer Science,* 74-83.

9. L.S. Heath (1987): Embedding outerplanar graphs in small books. *SIAM J. Algebr. Discr. Meth. 8,* to appear.

10. D.J. Muder (1985): Book embeddings of regular complete bipartite graphs. Typescript, The MITRE Corp.

11. J. Riordan (1958): *An Introduction to Combinatorial Analysis,* John Wiley and Sons, New York.

12. A.L. Rosenberg (1983): The DIOGENES approach to testable fault-tolerant arrays of processors. *IEEE Trans. Comp., C-32,* 902-910.

13. E. Stöhr (1987): A tradeoff between pagenumber and width of book embeddings of graphs. Typescript, East Germany Academy of Sciences.

14. M.M. Syslo (1979): Characterizations of outerplanar graphs. *Discr. Math. 26*, 47-53.

15. M. Yannakakis (1986): Four pages are necessary and sufficient for planar graphs. *17th ACM Symp. on Theory of Computing.*

Height	Arity					
	2	3	4	5	6	7
3	0.400	-	-		-	-
4	0.645	0.298	-			
5	0.794	0.511	0.176	-	-	-
6	0.882	0.664	0.330	0.092	-	-
7	0.933	0.772	0.467	0.190	0.045	
8	0.963	0.846	0.584	0.296	0.103	0.021
9	0.979	0.897	0.680	0.401	0.175	0.052
10	0.989	0.931	0.755	0.498	0.256	0.097
11	0.994	0.954	0.814	0.585	0.339	0.154
12	0.997	0.969	0.860	0.659	0.421	0.218
13	0.998	0.979	0.894	0.722	0.498	0.286
14	0.999	0.986	0.920	0.775	0.569	0.355
15	0.999	0.991	0.940	0.818	0.632	0.423
16	1.000	0.994	0.955	0.853	0.687	0.487
17	1.000	0.996	0.966	0.882	0.735	0.547
18	1.000	0.997	0.975	0.905	0.776	0.602
19	1.000	0.998	0.981	0.924	0.812	0.652
20	1.000	0.999	0.986	0.939	0.842	0.696

Table 1: **Lower bound on the probability that randomly chosen pivot node yields $d+4$ types.**

d-ary Trees															
Height	Arity														
	2			3			4			5			6		
	m	M	a	m	M	a	m	M	a	m	M	a	m	M	a
2	2	3	2.34	2	3	2.50	2	3	2.60	2	3	2.67	2	3	2.71
3	3	5	4.29	3	6	4.77	3	6	5.05	3	6	5.23	3	6	5.35
4	3	7	5.27	3	7	5.78	3	7	6.06	3	7	6.24	3	7	6.36
5	3	7	5.58	3	8	6.20	3	8	6.57	3	8	6.81	3	8	6.98
6	3	7	5.76	3	8	6.47	3	9	6.93	3	9	7.25	3	9	7.45
7	3	7	5.87	3	8	6.65	3	9	7.20	3	10	7.60	3	10	7.91
8	3	7	5.93	3	8	6.77	3	9	7.40	3	10	7.88	3	11	8.26
9	3	7	5.96	3	8	6.84	3	9	7.55	3	10	8.11	3	11	8.54
10	3	7	5.98	3	8	6.90	3	9	7.66	3	10	8.26	-	-	-
11	3	7	5.99	3	8	6.93	3	9	7.75	-	-	-	-	-	-
12	3	7	5.99	3	8	6.95	-	-	-	-	-	-	-	-	-
13	3	7	6.00	-	-	-	-	-	-	-	-	-	-	-	-

Table 2: **Minimum (m), maximum (M) and average (a) number of d-ary tree vertex-types.**

X-Trees			
Height	m	M	a
3	6	7	6.86
4	11	14	12.47
5	13	20	17.74
6	13	25	20.35
7	14	26	22.71

Table 3: **Minimum (m), maximum (M) and average (a) number of X-tree vertex-types.**