SIAM J. DISC. MATH.
Vol. 2, No. 2, pp. 145-155, May 1989

001

UNIVERSAL GRAPHS FOR BOUNDED-DEGREE TREES AND PLANAR GRAPHS*

SANDEEP N. BHATT†, F. R. K. CHUNG‡, F. T. LEIGHTON§, AND ARNOLD L. ROSENBERG¶

Abstract. How small can a graph be that contains as subgraphs all trees on n vertices with maximum degree d? In this paper, this question is answered by constructing such universal graphs that have n vertices and bounded degree (depending only on d). Universal graphs with n vertices and $O(n \log n)$ edges are also constructed that contain all bounded-degree planar graphs on n vertices as subgraphs. In general, it is shown that the minimum universal graph containing all bounded-degree graphs on n vertices with separators of size n^α has $O(n)$ edges if $\alpha < \frac{1}{2}$; $O(n \log n)$ edges if $\alpha = \frac{1}{2}$; $O(n^{2\alpha})$ edges if $\alpha > \frac{1}{2}$.

Key words. universal graphs, separators, trees, planar graphs

AMS(MOS) subject classifications. 05C, 65W, 68Q

1. Introduction. Given a family F of graphs, a graph G is said to be F-universal if G contains every graph in F as a subgraph. A fundamental problem of interest is to determine how few edges a universal graph can have. Such problems are of interest in circuit design [V], data representation [CRS], [RSS], and parallel computing [BLe], [BCLR].

Let $f(F)$ denote the minimum number of edges in a graph that contains all graphs in F as subgraphs. There is a large literature on universal graphs for various families of graphs. In the early 1960s, Rado first investigated universal graphs for infinite graphs [Ra]. Since then many results on this subject have been published. Here we give a list of some of the known results about universal graphs for various families of graphs.

(1) Moon [M] considered the universal graphs that contain all graphs on n vertices as induced subgraphs. He established upper and lower bounds for the number of vertices in such universal graphs.

(2) Bondy [Bo] investigated universal graphs for the class C_n of all cycles of length $\leqq n$; such universal graphs are called pancyclic and he showed that

$$n+\log_2(n-1)-1<f(C_n)<n+\log_2(n-1)+\log^* n+O(1)$$

where $\log^* n$ denotes

$$\min\{k: \overbrace{\log\log\cdots\log}^{k} n<2\}.$$

(3) Let T_n denote the class of all trees on n vertices. A lower bound of $\frac{1}{2}n\log n$ for $f(T_n)$ can be obtained by considering degree sequences of the universal graph. The upper

* Received by the editors July 1, 1988; accepted for publication (in revised form) October 24, 1988.

† Department of Computer Science, Yale University, New Haven, Connecticut 06520. The research of this author was supported in part by National Science Foundation grant MIP-86-01885.

‡ Bellcore, Morristown, New Jersey 07960.

§ Department of Mathematics and Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139. The research of this author was supported in part by U.S. Air Force contract OSR-86-0076, Defense Advanced Research Projects Agency (DARPA) contract N00014-80-C-0622, U.S. Army contract DAAL-03-86-K-0171, and a National Science Foundation Presidential Young Investigator Award with matching funds from AT&T and IBM.

¶ Department of Computer Science, University of Massachusetts, Amherst, Massachusetts 01003. The research of this author was supported by National Science Foundation grants DCI-85-04308 and DCI-87-96236.

bound was improved by a series of papers [CG1], [CG2], [CG3], [CGP] and the best known upper bound is only a constant multiple of the lower bound [CG3]

$$\frac{1}{2} n \log n \leqq f(T_n) \leqq \frac{7}{\log 4} n \log n + O(n).$$

(4) One variation on universal graph problems is to require the universal graph to satisfy specified properties. In [CCG], it was shown that a minimum tree that contains all trees on n vertices must have $n^{(1+o(1))\log n/\log 4}$ vertices, and that this is the best possible.

(5) A *caterpillar* is a tree with the property that its vertices of degree greater than one induce a path. Kimble and Schwenk [KS] first considered the problem of determining minimum caterpillars that contains all caterpillars on n vertices, and they gave some estimates of the size of the universal caterpillar. In [CGS], it was shown that the minimum number of edges in such a universal caterpillar is within a constant factor of $n^2/\log n$.

(6) Let E_n denote the class of all graphs with n edges. It turns out that E_n-universal graphs contain many more edges than T_n-universal graphs. In fact, it was shown in [BCEGS] that

$$\frac{cn^2}{\log^2 n} < f(E_n) < (1+o(1)) \frac{n^2 \log \log n}{\log n}.$$

In this paper, we consider universal graphs for the family $T_{n,d}$ of all trees on n vertices with maximum degree d. We construct $T_{n,d}$-universal graphs on n vertices with bounded degree (depending only on d). In related independent work, Friedman and Pippenger [FP] recently proved that an expander graph on cn vertices with constant degree contains all trees on n vertices with maximum degree d. (The constant c, which depends on d, is quite large.)

We will also consider universal graphs for the family $P_{n,d}$ of all planar graphs on n vertices with maximum degree d. The $P_{n,d}$-universal graphs have n vertices and $O(n \log n)$ edges, improving the previous bound of $O(n^{3/2})$ in [BCEGS].

In § 2, we use graph separators to construct universal graphs with $O(n)$ edges for the family of binary trees on n vertices. Using similar techniques, we derive universal graphs for families of graphs of bounded degree with small separators. In particular, we obtain universal graphs with $O(n \log n)$ edges for bounded-degree planar graphs on n vertices and universal graphs with $O(n)$ edges for bounded-degree outerplanar graphs on n vertices. We also obtain $T_{n,d}$-universal graphs on n vertices and $O(n)$ edges, but the maximum degree of these graphs is of order $O(\log n)$. To reduce the maximum degree, we modify our construction using expander graphs in § 3 and the resulting $T_{n,d}$-universal graphs have bounded degree. Section 4 concludes with further problems and remarks.

2. Universal graphs for families of graphs with small bisectors. In a graph G on n vertices, a set S of vertices is called a *bisector* if, by removing vertices in S from G, the remaining graph can be partitioned into two exactly equal parts so that there is no edge joining a vertex from one part to the other.

Here we need a stronger notion of bisectors, called *k-bisectors*. When the vertices of a graph G are colored in k colors, a set S of vertices is said to be a k-bisector if, by removing vertices in S from G, the remaining graph can be partitioned into two exactly equal parts so that each part contains exactly equal numbers of vertices of each color, and there is no edge joining a vertex from one part to the other. Any tree on n vertices has a bisector of size $c \log n$ and a k-bisector of size $ck \log n$ for some constant c [C], [LT], [BL]. For binary trees, $c = (\log 3)^{-1} < 1$.

We first consider a simpler version of our problem, namely, for binary trees, that are trees with maximum degree three.

LEMMA 1. *The n vertices of a binary tree T can be mapped into a complete binary tree C on no more than $2^q - 1$ vertices* ($2^q - 1 \leqq n < 2^{q+1} - 1$) *so that at most* 6 $\log(n/2^{t-3})$ *vertices*[1] *of T are mapped into a vertex of C at distance t from the root, and so that any two vertices adjacent in T are mapped to vertices at most three apart in C.*

Proof. The idea is to recursively bisect T, placing the successive sets of bisector vertices within successively lower levels of C, until T is decomposed into single vertices. For example, the vertices placed at the root of C bisect T into two subgraphs T_1 and T_2. Similarly, vertices mapped to the left child of the root bisect T_1, and vertices mapped onto the right child bisect T_2. In addition, at level i of C we map vertices of T (that have not already been mapped within levels $i-1$, $i-2$) that are adjacent to vertices mapped at level $i-3$ of C. This ensures that vertices adjacent in T will be mapped to vertices of C distance three apart.

To keep the number of vertices of T mapped to a level i vertex in C within the required bounds, we use 3-bisectors. The following procedure describes how this is done.

Step 0. Initialize every vertex of T to color A, bisect T, and place the bisector vertices at the root (level 0) of C.

Step 1. For each subgraph created in the previous step, recolor every vertex adjacent to the bisector in the previous step with color 0, and place a 2-color bisector for the subgraph at the corresponding level-1 vertex of C.

Step 2. For each subgraph created in the previous step, recolor every vertex of color A adjacent to the bisector in the previous step with color 1, and place a 3-color bisector for the subgraph at the corresponding level-2 vertex of C.

Step t ($\log |T| \geqq t \geqq 3$). For each subgraph created in the previous step, place every vertex of color $t-1 \pmod 2$ at the corresponding level t of C, recolor every vertex of color A that is adjacent to a vertex mapped at the previous level with color $t-1 \pmod 2$ and place a 3-color bisector for the remaining subgraph at the corresponding level t vertex of C.

To ensure the accuracy of Step t, it suffices to show $n_t \leqq 6\log(n/2^t) + 18$ for $3 \leqq t < \log |T|$. Since we have

$$\begin{aligned} n_t &\leqq 3\log\frac{n}{2^t} + \frac{1}{2}n_{t-3} \\ &\leqq 6\log\frac{n}{2^t} + 18, \end{aligned}$$

Lemma 1 is proved.

The analogous version for higher-degree trees and planar graphs can be proved in a very similar way, and the proofs are left to the reader. The main difference in proving these results is that vertices adjacent to previously mapped vertices are themselves only mapped at every $\theta(\log d)$ level instead of at every level. This way, only two or three colors are needed, and the bisector at every level stays small.

[1] Strictly speaking, we should use $\lfloor 6\log(n/2^{t-3})\rfloor$ instead of $6\log(n/2^{t-3})$. However, we will usually not bother with this type of detail since it has no significant effect on the arguments or results. Also, all logarithms henceforth are of base two.

LEMMA 2. *The vertices of a tree T with maximum degree d can be mapped into a complete binary tree C on no more than $2^q - 1$ vertices $(2^q - 1 \leqq n < 2^{q+1} - 1)$ so that $O(\log(n/2^t))$ vertices of T are mapped to a vertex of C at distance t from the root, and so that any two vertices adjacent in T are mapped to vertices at most distance $O(\log d)$* $2 \log d + 2$ *apart in C.*

LEMMA 3. *The vertices of a planar graph G of maximum degree d can be mapped into complete binary tree C on $2^q - 1$ vertices $(2^q - 1 \leqq n < 2^{q+1} - 1)$ so that $O(\sqrt{n/2^t})$ vertices of G are mapped to a vertex of C at distance t from the root, and so that any two vertices adjacent in G are mapped to vertices at most distance $O(\log d)$ apart in C.*

A graph G is said to have a *k-bisector function f* if any subgraph of G on m vertices has a k-bisector of size no more than $f(m)$. The preceding lemmas are all special cases of the following.

LEMMA 4. *Suppose G on n vertices with maximum degree d has a k-bisector function f. The vertices of G can be mapped into a complete binary tree C on no more than $2^q - 1$ vertices where $2^q - 1 \leqq n < 2^{q+1} - 1$ so that $O(f(n/2^t))$ vertices of G are mapped to a vertex of C at distance t from the root, and so that any two vertices adjacent in G are mapped to vertices at most distance $O(k \log d)$ apart in C if k is large enough that*

$$2f(xd^{3k}) \leqq d^{3k-4} f(x) \quad \text{for all } x.$$

Although Lemma 4 looks somewhat complicated, it is a natural generalization of Lemmas 1–3, and we omit the proof. We can now construct universal graphs using the decomposition lemmas.

THEOREM 1. *The minimum universal graph for the family of all bounded degree trees on n vertices has n vertices and $O(n)$ edges.*

Proof. Using Lemma 2, we consider the graph with vertices grouped into clusters corresponding to the vertices in the complete binary tree C. A cluster corresponding to a vertex of level t contains $O(\log(n/2^t))$ vertices. We connect all pairs of vertices in clusters with corresponding vertices within distance $O(\log d) = O(1)$ apart in C. By Lemma 2 the resulting graph is universal for the family of all trees with maximum degree d. The number $h(n)$ of edges in this graph is $O(n)$, since $h(n)$ satisfies the following recurrence inequality:

$$h(n) \leqq 2h\left(\frac{n}{2}\right) + c(\log n)^2$$

where c is an appropriate constant depending on d.

The construction just described has $O(n)$ vertices. To obtain a universal graph with precisely n vertices, we modify the embedding of Lemma 1 so that the same number of nodes of T are wrapped to nodes in the same level of C. This is easy to do since we can always arbitrarily expand the bisector of any subtree to be within one of its maximum allowed value (which is the lesser of the number of nodes remaining and $O(\log(n/2^t))$ for nodes on level t of C. The exact value of the maximum bisector is the same for all nodes on a level and depends on the parity of the number of nodes in the subgraphs at that level. Hence, the size of the bisectors at each level depends only on n, and the universal graph can be assumed to have precisely n nodes.

THEOREM 2. *The minimum universal graph for the family of all bounded-degree planar graphs on n vertices has n vertices and $O(n \log n)$ edges.*

Proof. The construction is obtained by using Lemma 3 in similar fashion as in the proof of Theorem 1. The number of edges $h(n)$ satisfies

$$h(n) \leqq 2h\left(\frac{n}{2}\right) + cn$$

and, therefore, the minimum universal graph has $O(n \log n)$ edges.

THEOREM 3. *The minimum universal graph for a family of bounded-degree graphs on n vertices with bisector function $f(x) = x^\alpha$ has n vertices with $O(n)$ edges if $\alpha < \frac{1}{2}$; $O(n \log n)$ edges if $\alpha = \frac{1}{2}$; $O(n^{2\alpha})$ edges if $\alpha > \frac{1}{2}$.*

Proof. The construction follows from Lemma 4 together with the fact that the number $h(n)$ of edges satisfies

$$h(n) \leqq 2h\left(\frac{n}{2}\right) + c(f(n))^2.$$

THEOREM 4. *The minimum universal graph for the family of all bounded-degree outer-planar graphs on n vertices has n vertices and $O(n)$ edges.*

Proof. Since an outerplanar graph on n vertices has a bisector of size $O(\log n)$, the result follows from Theorem 3.

3. A bounded-degree universal graph for bounded-degree trees. For the family of bounded-degree trees, the minimum universal graph has n vertices and $O(n)$ edges as indicated in Theorem 2; however, the maximum degree is of order $\log n$. Although the number of the edges in this universal graph is within a constant factor of the optimum, its vertices have unbounded degree.

In this section we describe a construction for graphs on n vertices with bounded-degree that are universal for all bounded degree trees on n vertices. First we need a few definitions.

DEFINITION. A graph $G(V, E)$ is said to be *full* if for every $V' \subset V$, $|V'| \leqq |V|/2$, the number of edges between V' and $V - V'$ is at least $|V'|$.

We observe that there is a constant δ such that for every m, there is a full graph on m vertices with maximum degree δ. Any expander graph can be used for constructing full graphs [AC], [LPS]. It was shown in [AC] that in any δ-regular graph $G(V, E)$ with second largest eigenvalue λ, for every $V' \subset V$ with $|V'| = \alpha n$, the number $e(V')$ of edges contained in V' satisfies

$$|e(V') - \tfrac{1}{2}\delta\alpha^2 n| \leqq \lambda\alpha(1-\alpha)n.$$

Therefore, there are at least $(\delta - 2\lambda)\alpha(1 - \alpha)n$ edges between V' and $V - V'$. As long as $(\delta - 2\lambda)/2 \geqq 1$, the graph G is full. For large enough δ, this is usually the case.

The universal graph H on n vertices is obtained as follows. For simplicity, we will assume that $n = 2^\alpha - 1$.

Start the construction with a complete binary tree on n vertices. Then, add edges so that the vertices at level k (a constant specified later) form a full graph on 2^k vertices. Repeat this for vertices at levels $2k, 3k, \cdots$. Call the resulting graph H_0.

Next, add extra edges so that the vertices at levels $k, 2k, \cdots, \log n - s$ (k divides $\log n - s$ and s is a constant specified later) collectively form a full graph. Call the resulting graph H_1.

Finally, insert an edge between any pair of vertices within distance t of each other, where t is a constant specified later. The resulting graph, denoted by H, is our universal graph. Observe that the maximum degree of any vertex in H is no greater than

$(2\delta + 3)^t$ which, of course, is a constant because δ and t are. We will show that H is universal for the family of bounded-degree trees on n vertices if k, s, and t are properly chosen.

THEOREM 5. *For the family of trees on n vertices with maximum degree d, we can construct universal graphs on n vertices with bounded degree (that depends only on d).*

The proof of Theorem 5 is somewhat involved, and requires a few combinatorial facts concerning full graphs and trees. The intuition captured in the following lemmas may be understood as follows. Suppose that we have mapped a subset of the vertices of a tree T within a graph G, and we next wish to map a vertex v of T onto a vertex of G in such a way that it remains "close to" its neighbors that have already been embedded. If there is no place readily available, we can still find a suitable place for v by "perturbing" the existing mapping slightly to make room for v. The "flow lemmas" establish conditions under which this can be done without dilating edges significantly.

LEMMA 5. *Let G be a full graph with maximum degree d, and consider any assignment of packets to vertices of G such that every vertex of G is assigned at least $\lceil d/2 \rceil$ packets. Then, for any disjoint subsets S and T of vertices such that $|S| = |T|$, it is possible to redistribute the packets so that we have the following*:

(i) *Every packet either stays stationary or moves to a neighbor in G*;

(ii) *The number of packets in each vertex in S decreases by one*;

(iii) *The number of packets in each vertex in T increases by one*; *and*

(iv) *The number of packets in each vertex in $V - (T \cup S)$ remains the same.*

Proof. The lemma is proved with a simple max-flow/min-cut argument. Set up a flow problem with a supersource connected to each vertex in S and a supersink connected to each vertex in T. Assign unit capacity to each edge. Because G is full, there is a 0–1 flow with value $|S|$ between the source and sink. The flow determines a one-to-one correspondence (along with edge-disjoint paths) from the vertices in S to the vertices in T. By moving one packet forward along each edge that has unit flow we can effect a reassignment of packets that satisfies conditions (ii)–(iv).

Since every vertex in the flow graph (with the supersource and supersink) has degree at most $d + 1$, at most $\lfloor (d + 1)/2 \rfloor = \lceil d/2 \rceil$ packets will be removed from any vertex of G during the reassignment process. Since every vertex of G initially has $\lceil d/2 \rceil$ packets, no packet need ever move more than one step. Hence, the reassignment also satisfies the first condition.

LEMMA 6. *Let G be a full graph on n vertices with maximum degree d, and consider any assignment of packets to vertices of G so that vertex v_i has a_i packets, where $a_i \geqq \lceil d/2 \rceil$ for $1 \leqq i \leqq m$. Then for any set of numbers $\{a'_i \mid 1 \leqq i \leqq m\}$ for which $a'_i \geqq \lceil d/2 \rceil$ for $1 \leqq i \leqq m$, it is possible to redistribute the packets so that we have the following*:

(i) *Every packet is reassigned to a vertex that is at distance at most* $\max_{1 \leqq i \leqq m} |a_i - a'_i|$ *from its original location in G*; *and*

(ii) *The number of packets assigned to v_i changes from a_i to a'_i, for all $1 \leqq i \leqq m$.*

Proof. Apply Lemma 5 for $\max_{1 \leqq i \leqq m} |a_i - a'_i|$ iterations, each iteration decreasing the maximum value of $|a_i - a'_i|$, $1 \leqq i \leqq m$, by one.

To establish Theorem 5 we use a decomposition strategy different from that in § 2. The following lemma is a simple extension of the $\frac{1}{3} : \frac{2}{3}$ separator theorem for binary trees and was observed previously in [BLe]. This can be generalized to arbitrary maximum degree d via the $(1/(d + 1), d/(d + 1))$ separator theorem (see [C]).

LEMMA 7. *For every constant $p < \frac{1}{2}$, there exists a constant q such that any n-vertex two-colored binary forest with w vertices of color A can be partitioned into two sets by the* *removal of q edges so that each set has at least $\lceil pn \rceil$ vertices and at least $\lceil pw \rceil$ vertices of color A.*

We also require an additional, final lemma.

LEMMA 8. *Every binary tree T on n vertices can be embedded within H_0 so that we have the following:*

(i) *Vertices of T are assigned to vertices in levels $0, k, 2k, \cdots, \log n - s$ of H_0;*

(ii) *Every vertex in levels $0, k, 2k, \cdots, \log n - s$ of H_0 is assigned at least $\lceil d/2 \rceil$ and at most c_1 vertices of T, where c_1 is some constant;*

(iii) *Vertices adjacent in T are assigned to vertices in H_0 separated by distance at most c_2, for some constant c_2.*

Once Lemma 8 is established, it is easy to complete the proof of Theorem 5 as follows.

Proof of Theorem 5. First obtain the embedding of Lemma 8. Next, by Lemma 6 we can use the edges of $H_1 - H_0$ to reassign the vertices of T within H_1 so that we have the following:

(i) Every vertex in levels $0, k, 2k, \cdots, \log n - s - k$ of H_1 is assigned $2^k - 1$ vertices of T;

(ii) Every vertex in level $\log n - s$ of H_1 is assigned $2^s - 1$ vertices of T; and

(iii) Vertices adjacent in T are assigned to vertices in H_1 separated by distance at most c_3, where $c_3 \leqq c_2 + 2 \max (|2^s - 1 - \lceil d/2 \rceil|, |2^k - 1 - \lceil d/2 \rceil|)$.

At this point, we need only require that $s \geqq k$ and that $2^k - 1 \geqq \lceil d/2 \rceil$ so that the conditions of Lemma 6 are satisfied. Since k, s, d, c_1, and c_2 are all constants, we know that c_3 also is constant. We now reassign vertices one more time so that the mapping from T to H becomes one-to-one and onto. This is done by arbitrarily assigning the vertices of T on levels $0, k, 2k, \cdots, \log n - s$ of H_1 to their immediate descendants. Once this is done, the maximum distance in H_1 between any two nodes adjacent in T will be at most $c_3 + 2s$, which is constant. By setting $t = c_3 + 2s$ in the construction of H, this will mean that T is a subgraph of H, thereby completing the proof of Theorem 5.

Proof of Lemma 8. We follow an approach similar to that in § 2. However, since we are allowed to place only $O(1)$ vertices of T at any one vertex of H_0, we cannot afford to bisect the tree at each step because that may require placing $c \log n$ vertices of T at the root of H_0 for some constant c. Therefore, instead of bisecting the tree at each step, we separate it into proportional size components using Lemma 7, and continually balance the sizes of components as the embedding proceeds towards lower levels of H_0.

Initially, color all the vertices of T white. Then, pick any $\lceil d/2 \rceil$ vertices of T and map them to the root (level 0) of H_0. Color *red* those vertices of T that are adjacent to one or more of the vertices placed at the root of H_0. Next, fix p with $\frac{1}{3} \leqq p < \frac{1}{2}$, and use Lemma 7 to partition the (as yet unmapped) vertices of T into two sets, each with at least the fraction p of the total number of unmapped vertices, and each with at least the fraction p of the total number of red vertices (always rounded up to the nearest integer, of course). One of the sets is distributed to the left subtree of the root of H_0 and the other set to the right subtree. By Lemma 7, no more than q edges connect vertices in the two sets.

No vertices of T will be assigned to the next $k - 1$ levels of H_0, but we continue to partition T into smaller and smaller sets. In particular, we first color vertices in the "left set" of T (those unmapped vertices of T assigned to the left subtree of H_0) that are adjacent to vertices in the right set. We then use Lemma 7 to partition the left and right sets each into two smaller subsets, one for each grandchild of the root. Continue in this fashion, coloring vertices red as they become adjacent to vertices in the opposite set and splitting the forests (sets) into smaller forests until we have distributed a forest to each vertex on the kth level of H_0.

Although the vertices are split into roughly equal proportions ($p : 1 - p$) at each level, the sizes of forests at the kth level could vary substantially (in fact, anywhere between p^k and $(1 - p)^k$). Therefore, at this stage we balance the sizes of the forests assigned to each vertex by redistributing forests among vertices at level k. To achieve this balance, first use Lemma 7 to partition each forest into $\lceil d/2 \rceil$ subforests (but we do not distribute the subforests further down the tree). Next, we partition each subforest whose size is greater than $1/p$ times the size of the smallest subforest. Observe that this does not affect the size of the smallest subforest.

We are now ready to apply Lemma 6, with each subforest represented as a packet. In particular, we use Lemma 6 to redistribute subforests on the level so that every vertex ends up with an equal number of subforests (to within one). We then map all the red vertices of T (i.e., those adjacent to vertices in different subforests) to the corresponding vertex of H_0 where the enclosing subforest is currently located, making sure to map at least $\lceil d/2 \rceil$ vertices of T to each vertex on level k in H_0. (If there are not enough red vertices, then we use some of the white vertices in the same subforest to make up the total. We show later that there are always enough vertices overall so that this is possible.)

After the mapping is completed for level k, we recolor red all white vertices of T that are adjacent to vertices already mapped, and we henceforth regard the collection of subforests assembled at a single vertex of H_0 as a single forest. Next, we repeat the process used on levels $1, 2, \cdots, k$ for levels $k + 1, k + 2, \cdots, 2k, \cdots, \log n - s$, where s is a constant yet to be specified. At every kth level, we rebalance and coalesce forests as on level k, and map all red vertices of T to the corresponding vertices of H_0. At level $\log n - s$ all the unmapped vertices of T (both red and white) are mapped directly to the corresponding vertices of H_0. Several details remain to be ironed out; however, it should be clear that vertices adjacent in T are mapped to vertices which are at most k levels apart in H_0.

The analysis needed to complete the proof is tedious, but not difficult. We start by letting r_{ik} be the maximum number of red vertices in any forest after all partitioning, balancing, coalescing, mapping, and recoloring is done at level ik of H_0. Similarly, let z_{ik} be the number of vertices (both red and white) in the smallest forest at level ik.

We will prove by induction that, for $ik \leqq \log n - s$, $z_{ik} \geqq 2^{-ik}n/6$, and $r_{ik} \leqq r' = 96(1 + q)2^{-k}p^{-(k+\lceil \log \lceil d/2 \rceil \rceil + 1)}$.

Observing that $r' \geqq d\lceil d/2 \rceil$ for k sufficiently large (in terms of p and d), we note that both statements are trivially true for $i = 0$ and n sufficiently large. We next calculate bounds for r_{ik+k} and z_{ik+k} to proceed with the inductive step.

By Lemma 7, we know that

$$r_{ik+1} \leqq (1 - p) r_{ik} + 1 + q;$$

therefore, each forest at level $ik + k$ of H_0 has at most $(1 - p)^k r_{ik} + (1 + q)/p$ red vertices initially. The process of partitioning forests into subforests at level $ik + k$ cannot increase this value, but redistributing, coalescing, and recoloring certainly can. To measure their effect, we need to bound the number of subforests that are located at any vertex following redistribution. This of course depends on the overall number of subforests, which in turn depends on the size of the smallest subforest.

The size of the smallest subforest at level ik is z_{ik}. Hence, the size of the smallest forest at level $ik + 1$ is at least $pz_{ik} - 1$. Applying the argument recursively, we find that the size of the smallest subforest at level $ik + k$ (after all the subdividing at this level is complete) is, for $p \leqq \frac{1}{2}$, at least

$$z_{ik}p^{k+\lceil\log\lceil d/2\rceil\rceil} - (1-p)^{-1} \geqq p^{k+\lceil\log\lceil d/2\rceil\rceil}2^{-ik}n/6-2.$$

For sufficiently large s (i.e., small enough i), this is at least $p^{k+\lceil\log\lceil d/2\rceil\rceil}2^{-ik}n/12$. Hence, the number of subforests at this level is no greater than $12 \times 2^{ik}p^{-(k+\lceil\log\lceil d/2\rceil\rceil)}$. The maximum number of subforests located at any vertex after balancing is, therefore, no greater than

$$1+12\times 2^{-k}p^{-(k+\lceil\log\lceil d/2\rceil\rceil)} \leqq 24\times 2^{-k}p^{-(k+\lceil\log\lceil d/2\rceil\rceil)}.$$

Consequently, the maximum number of red vertices in any forest after rebalancing and coalescing is at most

$$((1-p)^k r_{ik} + (1+q)/p)24\times 2^{-k}p^{-(k+\lceil\log\lceil d/2\rceil\rceil)}.$$

Since mapping and recoloring can increase this at most by a factor of two, we have

$$r_{ik+k} \leqq 48(1-p)^k r_{ik}2^{-k}p^{-(k+\lceil\log\lceil d/2\rceil\rceil)} + 48(1+q)2^{-k}p^{-(k+\lceil\log\lceil d/2\rceil\rceil)}.$$

By choosing $p > \frac{1}{3}$ so that $(1-p)/2p < 1$, we have that for k sufficiently large (in terms of p and d):

$$r_{ik+k} \leqq \tfrac{1}{2}r_{ik} + 48(1+q)2^{-k}p^{-(k+1+\lceil\log\lceil d/2\rceil\rceil)};$$

thus,

$$r_{ik+k} \leqq 96(1+q)2^{-k}p^{-(k+1+\lceil\log\lceil d/2\rceil\rceil)} = r',$$

as claimed.

We next complete the inductive step for z_{ik+k}. Since the largest and smallest subforests differ in size by at most a factor of $1/p$, the size of the smallest forest after balancing and coalescing is at least $p(n - r'2^{ik+k})2^{-(ik+k)-1}$, the factor $\frac{1}{2}$ accounting for the fact that every vertex has the same number of packets to within one. After mapping and recoloring, the size of the smallest forest is

$$z_{ik+k} \geqq \frac{p}{2}(n - r'2^{ik+k})2^{-(ik+k)} - r'.$$

With some additional calculations it can be checked that this is at least $2^{-(ik+k)}n/6$ for $p > \frac{1}{3}$ and s sufficiently large, thereby completing the proof of the claim.

By choosing s sufficiently large, we have shown that every vertex at levels $0, k, \cdots, \log n - s - k$ of H_0 is assigned at least $\lceil d/2\rceil$ and at most r' vertices of T. Since s is constant, every vertex at level $\log n - s$ of H_0 is assigned between $\lceil d/2\rceil$ and c_1 vertices, where c_1 is some constant bigger than r'. Moreover, vertices of T are assigned only to vertices in levels $0, k, \cdots, \log n - s$ of H_0. Hence, it remains only to show that vertices adjacent in T are assigned to vertices in H_0 that are separated by distance at most c_2, for some constant c_2. We already know that c_2 is at most k plus the distance subforests are allowed to move during the rebalancing step at every kth level. By Lemma 6, this distance is at most the largest number of subforests at any vertex before rebalancing. By the construction, this is at most some constant determined by p, d, k, and s. This completes the proof of Lemma 8 and Theorem 5.

4. Further problems and remarks. While the universal graph for bounded-degree trees is optimal (within a constant factor), the universal graph for bounded-degree planar graphs on n vertices has $O(n \log n)$ edges. On the other hand, the best known lower bound for the number of edges is still cn. It is of interest to close up the gap.

One variation of the universal graph problem is to require the universal graph to be of some specified type. For example, in [CCG] universal trees that contain all trees with at most n nodes were considered. Relatively little is known about universal planar graphs that contain all planar graphs on n vertices.

This work is heavily motivated by simulation of graph families in various host networks with small *dilation* (i.e., adjacent vertices are mapped into nearby vertices) and small *expansion* (i.e., the ratio of the size of host graph and the maximum size of graphs in the family is small). The decomposition lemma (Lemma 1 in § 2) for binary trees also provides optimal embeddings of binary trees within other structures. For example, we can show that every n-vertex binary tree can be embedded within an n-vertex complete binary tree with expansion 1 and dilation $O(\log \log n)$. This settles a conjecture of Hong, Mehlhorn, and Rosenberg [HMR] who showed a lower bound of $\Omega(\log \log n)$ for this problem. By embedding a complete binary tree within the shuffle exchange graph with expansion 1 and dilation 2, we obtain $O(\log \log n)$ dilation for arbitrary trees embedded within the shuffle-exchange graphs. Similarly, we have recently shown that an n-vertex binary tree can be embedded with constant expansion and dilation within the butterfly network [BCHLR]. Finally, we have shown that all binary trees can be embedded in a hypercube with expansion 1 and dilation 10 [BCLR].

REFERENCES

[AC] N. Alon and F. R. K. Chung, *Explicit construction of linear sized tolerant networks*, Discrete Mathematics, 72 (1988), pp. 15–20.

[BCEGS] L. Babai, F. R. K. Chung, P. Erdös, R. L. Graham, and J. Spencer, *On graphs which contain all sparse graphs*, Ann. Discrete Math., 12 (1982), pp. 21–26.

[BL] S. N. Bhatt and F. T. Leighton, *A framework for solving* VLSI *graph layout problems*, J. Comput. Systems Sci., 28 (1984), pp. 300–343.

[BLe] S. N. Bhatt and C. E. Leiserson, *How to assemble tree machines*, in Advances in Computing Research 2, F. Preparata, ed., JAI Press, New York, 1984.

[BCLR] S. N. Bhatt, F. R. K. Chung, T. Leighton, and A. L. Rosenberg, *Optimal simulations of tree machines*, in Proc. 27th Annual IEEE Symposium on Foundations of Computer Science, Toronto, 1986, pp. 274–282.

[BCHLR] S. N. Bhatt, F. R. K. Chung, J.-W. Hong, T. Leighton, and A. L. Rosenberg, *Optimal simulations by butterfly networks*, in Proc. 20th Annual ACM Symposium on Theory of Computing, Chicago, 1988, pp. 192–204.

[Bo] J. A. Bondy, *Pancyclic graphs*, J. Combin. Theory Ser. B, 11 (1971), pp. 80–84.

[C] F. R. K. Chung, *Separator theorems and their applications*, to appear.

[CCG] F. R. K. Chung, D. Coppersmith, and R. L. Graham, *On trees which contain all small trees*, in The Theory of Applications of Graphs, G. Chartrand, ed., John Wiley, New York, 1981, pp. 265–272.

[CG1] F. R. K. Chung and R. L. Graham, *On graphs which contain all small trees*, J. Combin. Theory Ser. B, 24 (1978), pp. 14–23.

[CG2] ———, *On universal graphs*, Ann. New York Acad. Sci., 319 (1979), pp. 136–140.

[CG3] ———, *On universal graphs for spanning trees*, Proc. London Math. Soc., 27 (1983), pp. 203–211.

[CGP] F. R. K. Chung, R. L. Graham, and N. Pippenger, *On graphs which contain all small trees* II, in Proc. 1976 Hungarian Colloquium on Combinatorics, North Holland, Amsterdam, 1978, pp. 213–223.

[CGS] F. R. K. Chung, R. L. Graham, and J. Shearer, *Universal caterpillars*, J. Combin. Theory Ser. B, 31 (1981), pp. 348–355.

[CRS] F. R. K. Chung, A. L. Rosenberg, and L. Snyder, *Perfect storage representations for families of data structures*, SIAM J. Algebraic Discrete Math., 4 (1983), pp. 548–565.

[FP] J. FRIEDMAN AND N. PIPPENGER, *Expanding graphs contain all small trees*, Combinatorica, 7 (1987), pp. 71–76.

[HMR] J.-W. HONG, K. MEHLHORN, AND A. L. ROSENBERG, *Cost trade-offs in graph embeddings with applications*, Assoc. Comput. Mach., 30 (1983), pp. 709–728.

[KS] R. J. KIMBLE AND A. J. SCHWENK, *On universal caterpillars*, in The Theory and Applications of Graphs, Gary Chartrand, Yousef Alavi, Donald L. Goldsmith, Linda Lesniak, and Don R. Lick, eds., John Wiley, New York, 1981, pp. 437–447.

[LPS] A. LUBOTZKY, R. PHILLIPS, AND P. SARNAK, *Ramanujan graphs*, Combinatorica, 8 (1988) pp. 261–278.

[LT] R. J. LIPTON AND R. E. TARJAN, *A separator theorem for planar graphs*, SIAM J. Appl. Math., 36 (1979), pp. 177–189.

[M] J. W. MOON, *On minimal n-universal graphs*, Proc. Glasgow Math. Soc., 7 (1965), pp. 32–33.

[Ra] R. RADO, *Universal graphs and universal functions*, Acta Arith., 9 (1964), pp. 331–340.

[Ro] A. L. ROSENBERG, *Issues in the study of graph embeddings*, Lecture Notes in Computer Science, Springer-Verlag, Berlin, New York, 1981, pp. 150–176.

[RSS] A. L. ROSENBERG, L. J. STOCKMEYER, AND L. SNYDER, *Uniform data encodings*, Theoret. Comput. Sci., 2 (1980), pp. 145–165.

[V] L. G. VALIANT, *Universality considerations in* VLSI *circuits*, IEEE Trans. Comput., C-30 (1981), pp. 135–140.