INTERVAL HYPERGRAPHS

Arnold L. Rosenberg*

COINS Technical Report 87-86

*This reseach was supported in part by NSF Grants DCI-85-04308 and DCI-87-96236

INTERVAL HYPERGRAPHS

*Arnold L. Rosenberg**
Department of Computer and Information Science
University of Massachusetts
Amherst, MA 01003

August 28, 1987

Abstract

An n-vertex *interval hypergraph* (I-hypergraph, for short) I comprises the set $V_n = \{1, 2, \cdots, n\}$ of vertices and a multiset $E(I)$ of hyperedges, each of the form $\{k, k+1, \cdots, k+r\}$ $(k \geq 1,\ 1 \leq r \leq n-k)$. One can decide in linear time whether or not a given hypergraph is isomorphic to an I-hypergraph. The *size* of an I-hypergraph is the sum of the cardinalities of its hyperedges. An *embedding* of the graph $G = (V, E)$ in the n-vertex I-hypergraph I comprises an injection $\mu_v : V \rightarrow V_n$, and an injection $\mu_e : E \rightarrow E(I)$, satisfying: for all $(u, v) \in E$, $\{\mu_v(u), \mu_v(v)\} \subseteq \mu_e(u, v)$. The problem of finding the smallest I-hypergraph in which a given graph can be embedded is *NP*-complete; so also is the problem of whether or not a given graph G is embeddable in a given I-hypergraph I. Certain problems that are *NP*-complete for arbitrary hypergraphs are solvable in polynomial time for I-hypergraphs. Included here are the problem of hypergraph 2-colorability and the problem of hypergraph hamiltonianicity. Say that the finite family Γ has an α-separator $(1/2 \leq \alpha < 1)$ of size $S(n)$. Every m-vertex graph in Γ can be embedded in an I-hypergraph of size

$$m \cdot \left(\sum_{i=0}^{\lambda(m)} S(\alpha^i m) \right),$$

where $\lambda(M) =_{\text{def}} \log_{1/\alpha} M$. The n-vertex I-hypergraph I is *strongly universal* for the finite family of graphs Γ if: given any $W \subseteq V_n$ and any graph $G = (V, E) \in \Gamma$ with $|V| \leq |W|$, there is an embedding of G in I such that $\mu_v(V) \subseteq W$. There is an I-hypergraph of size

$$m \cdot \left(\sum_{k=1}^{\log m} \sum_{i=0}^{\lambda(2^k)} S(\alpha^i 2^k) \right),$$

that is strongly universal for Γ: m is the largest number of vertices in any $G \in \Gamma$. For many families Γ, including binary trees and any family for which $S(n)$ is of the form n^δ, these strongly universal I-hypergraphs are within a constant factor of smallest possible.

*This research was supported in part by NSF Grants DCI-85-04308 and DCI-87-96236.

1. INTRODUCTION

This paper combines the research topics of two families of investigations that have appeared in the literature in recent years.

The first type of investigation expands the object of study in graph-embedding research from graphs to hypergraphs, motivated by the popularity of "bus-oriented" architectures in present-day microelectronics. Three examples are:

- In [4], Bhatt and Leiserson construct, for each integer n, what we are calling an interval-hypergraph in which every n-vertex binary tree can be embedded;
- in [14], Peterson and Ting determine (among other things) the minimum size of an interval-hypergraph in which the complete graph K_n can be embedded;
- in [19], Stout studies mesh-structured processor arrays with busses as well as point-to-point communication links; he concludes (among other things) that "contiguous" busses are best from physical considerations.

The second type of investigation seeks, for a given finite family of graphs Γ, a graph $G(\Gamma)$ that is ***strongly universal*** for Γ in the sense of containing each graph in Γ as a subgraph, even if some positive fraction of the vertices of $G(\Gamma)$ are "killed".

- In [2], Beck establishes the existence of an $O(n)$-vertex $O(n)$-edge graph that remains universal for ($\leq n$)-vertex path graphs[1], even after some arbitrary positive fraction of the graph's vertices are killed;
- in [1], Alon and Chung present an explicit construction for the graphs advertised in [2];
- in [3], Beck studies the analogous problem for trees.
- in [7], Friedman and Pippenger show that expanding graphs contain all small trees, even after one kills a positive fraction of the vertices.

Some of the motivation for these studies is fault tolerance in arrays of processors.

In this paper, we are motivated by two issues concerning bus-oriented parallel architectures. First we wish to compare the properties of bus-oriented communication structures (as idealized by hypergraphs) as opposed to point-to-point communication structures (as idealized by graphs). Second, we wish to study the use of busses in designing fault-tolerant arrays of identical processors in an environment of VLSI (Very Large Scale Integrated) circuitry. To these ends, we formalize the notion of ***interval hypergraph*** studied informally

[1]The n-vertex path-graph P_n has vertices $\{1, 2, \cdots, n\}$ and edges $\{(i, i+1) \mid 1 \leq i < n\}$.

in [4, 14, 16, 17], we study a number of the basic properties of interval hypergraphs, and we seek small interval hypergraphs that are *strongly universal* for given finite families of graphs, in the sense described above. The main result of our study is an algorithm that produces these small strongly universal interval hypergraphs. Included among the results we establish here are the following:

- The recognition problem for interval hypergraphs is solvable in linear time.
- The problem of finding the smallest interval hypergraph in which a given graph can be embedded is NP-complete.
- The problem of deciding whether or not a given graph can be embedded in a given interval hypergraph is NP-complete.
- Certain problems, such as 2-colorability and hamiltonianicity that are NP-hard for general hypergraphs can be solved efficiently for interval hypergraphs.

We also make explicit the relationship between interval hypergraphs and the better-known interval graphs, both being characterized in terms of matrices with the so-called consecutive ones property.

Our main algorithm takes as input a finite family of graphs Γ and an α-separator function $S(n)$ for Γ ($1/2 \leq \alpha < 1$). The algorithm produces a strongly universal interval hypergraph $I(\Gamma)$ for Γ, of size (measured by the sum of the cardinalities of its hyperedges)[2]

$$SIZE(I(\Gamma)) = m \cdot \left(\sum_{k=1}^{\log m} \sum_{i=0}^{\lambda(2^k)} S(\alpha^i 2^k) \right),$$

where m is the number of vertices in the largest graph in Γ, and $\lambda(M) =_{\text{def}} \log_{1/\alpha} M$. For many families Γ, including binary trees and any family for which $S(n)$ is of the form n^δ, the interval hypergraphs $I(\Gamma)$ are optimal in $SIZE$ (to within a constant factor). Moreover, for such $S(n)$, the $SIZE$ of $I(\Gamma)$, which can be viewed as measuring the area required to lay $I(\Gamma)$ out in the plane, is just a small constant factor greater than the area of any "collinear" layout in the plane of the largest graph in Γ ("collinearity" demanding that the graph's vertices lie along a line) .

2. BACKGROUND

2.1. The Formal Framework

We define the various notions that underlie the objects we study and the techniques we bring to bear on them.

[2]All logarithms are to the base 2 unless indicated otherwise.

(Hyper)Graphs. A *hypergraph* $H = (V, E)$ comprises a set V of *vertices* and a multiset E of subsets of V, called *hyperedges*. A *graph* $G = (V, E)$ is a hypergraph for which each $e \in E$ is a doubleton; each such hyperedge is called an *edge*. In this paper, we consider only graphs that are *simple* in the sense that E is a *set*. All families of graphs considered here are finite.

Interval Hypergraphs. An n-vertex *interval hypergraph* (*I-hypergraph*, for short) I is a hypergraph whose vertices comprise the set $V_n = \{1, 2, \cdots, n\}$ and whose hyperedges all have the form $\{k, k+1, \cdots, k+r\}$ for some $k \geq 1$ and $1 \leq r \leq n-k$.

Letting G ambiguously denote a graph or a hypergraph, we denote by $|G|$ the number of vertices of the (hyper)graph G and by *SIZE*(G) the sum of the cardinalities of G's (hyper)edges.

Embedding. An *embedding* of the graph $G = (V_g, E_g)$ in the hypergraph $H = (V_h, E_h)$ comprises one-to-one mappings

$$\mu_v : V_g \rightarrow V_h \quad \text{and} \quad \mu_e : E_g \rightarrow E_h$$

such that, for each edge $(u, v) \in E_g$, the image vertices $\mu_v(u)$ and $\mu_v(v)$ are both elements of the image hyperedge $\mu_e(u, v)$; symbolically, $\{\mu_v(u), \mu_v(v)\} \subseteq \mu_e(u, v)$. We say that a hypergraph *contains* any graph that is embeddable in it.

Strong Universality. Let Γ be a finite family of graphs. The hypergraph $H = (V_h, E_h)$ is *strongly universal* for Γ if, given any set $W \subseteq V_h$: for every graph $G = (V_g, E_g)$ in Γ for which $|G| \leq |W|$,[3] there is an embedding of G in H with $\mu_v(V_g) \subseteq W$.

Graph Separators and Separation Profiles. Let α be a rational number in the range $1/2 \leq \alpha < 1$, and let $S(n)$ be a nondecreasing integer-valued function. The graph G has an *α-separator of size* $S(n)$ either if $|G| < 2$ or if the following holds: By removing at most $S(|G|)$ edges from G, one can partition G into subgraphs G_1 and G_2, each of size

$$\lfloor (1-\alpha)|G| \rfloor \leq |G_i| \leq \lceil \alpha |G| \rceil,$$

and each having an α-separator of size $S(n)$. A family of graphs Γ has an α-separator of size $S(n)$ iff each graph $G \in \Gamma$ does.

> It is easy to verify that the family of path-graphs has a (1/2)-separator $S_{path}(n) \equiv 1$; Valiant [20] has shown that the family of binary trees has a (2/3)-separator of size $S_{tree}(n) \equiv 1$; it is an immediate consequence of the Planar Separator Theorem [11] that the family of rectangular meshes has a (2/3)-separator of size $S_{grid}(n) = \sqrt{8n}$. We use α-separators here, rather than the more commonly used bisectors, since for certain families of graphs (e.g., binary trees), choices of α other than $\alpha = 1/2$ lead to strongly universal I-hypergraphs that are more *SIZE*-efficient by a logarithmic factor.

[3] We denote by $|S|$ the cardinality of the set S.

Let G be a graph, and let l be any integer $\geq \log_{1/\alpha}|G|$. The graph G has an *α-separation profile* (*α-SP*, for short)

$$\langle s_l, s_{l-1}, \cdots, s_1 \rangle,$$

each s_i a nonnegative integer, precisely if: by removing at most s_l edges from G, one can partition G into subgraphs G_1 and G_2, each of size $\leq \alpha|G|$, and each having an α-SP

$$\langle s_{l-1}, s_{l-2}, \cdots, s_1 \rangle.$$

Another view of separation profiles is given by the notion of a

$$\langle s_l, s_{l-1}, \cdots, s_1 \rangle\text{-}\textit{decomposition tree}$$

for G: If one has a graph G with an α-SP

$$\langle s_l, s_{l-1}, \cdots, s_1 \rangle,$$

then one can construct a depth-l binary tree whose root is G, and whose left and right subtrees are, respectively, the $\langle s_{l-1}, s_{l-2}, \cdots, s_1 \rangle$-decomposition trees of the graphs G_1 and G_2 mentioned above.

The notions "separator" and "separation profile" converge in the fact that every graph G having an α-separator of size $S(n)$ admits an α-SP

$$\langle s_l, s_{l-1}, \cdots, s_1 \rangle,$$

where each $s_i = S(\alpha^{l-i}|G|)$. We leave to the reader the task of translating this correspondence into a decomposition tree for G. By dint of this relationship, we may refer freely to the *$S(n)$-decomposition tree* of any graph having an α-separator of size $S(n)$.

2.2. The Intended Interpretation

In our motivating scenario, the graph G represents a *logical* array: its vertices represent the processors of the array, and its edges represent communication links interconnecting the processors. (Thus, in G, interprocessor communication is "point-to-point"). The I-hypergraph I represents a *physical* array we shall use to realize G: its vertices represent the processors of the array, and its hyperedges represent busses that the processors tap into, in order to realize the edges of the array. (Thus, in I, interprocessor communication is along busses.) A processor (vertex) can tap into any bus (hyperedge) it belongs to. *SIZE*(I) approximates the area required to lay I out in the plane (on a chip), using the following groundrules. The vertices of I get laid out in a row, in natural order; the hyperedges get run as busses above the row, with vertical wires connecting each processor/vertex to the hyperedges it belongs to. Busses and wires have unit width; vertices occupy side-s squares, where s is large enough for the vertex to have its complement of incident edges. Busses and wires are allowed to cross – at most two crossing at a point – but not to overlap in any other way.

With the above representation in mind, the mapping μ_v can be viewed as assigning logical processors to physical processors, while μ_e assigns communication links to the busses that will realize them. The compatibility condition assures that any pair of processors that are supposed to use a bus can both be connected to it; the one-to-one condition assures that a hyperedge is used to realize at most one edge, modelling our assumption that each bus is dedicated to a single link.

When discussing fault tolerance, the set $W \subseteq V_h$ are the operational processors, while those in $W - V_h$ have failed; one wants to realize the array G on the good processors of I.

2.3. Related Work

Aside from the motivating sources cited earlier, the research in this paper builds on the studies of I-hypergraphs in the following papers. In [4], one finds a construction of an n-vertex I-hypergraph in which one can embed any n-node binary tree. In [13], one finds lower bounds of the *SIZE* of an I-hypergraph that contains the complete graph K_n, even when the notion of embedding is generalized so that the mapping μ_e need not be one-to-one (one uses time-sharing to reconcile contention for the busses). In [19], families of I-hypergraphs are presented, that are optimal in the sense of being able to simulate all other hypergraphs efficiently. In [16, 17, 6], our strong universality problem was first enunciated; *SIZE*-optimal strongly universal I-hypergraphs were presented for any finite family of path-graphs or of binary trees. Indeed, Section 4 of the present paper extends to families of arbitrary graphs the ideas and techniques of the last three cited papers.

3. BASIC PROPERTIES OF I-HYPERGRAPHS

We present in this section a number of basic properties of I-hypergraphs. Either the properties or their proofs will establish connections between I-hypergraphs and other structures that are better known.

3.1. Recognizing I-Hypergraphs

Our first result considers the problem of recognizing when a given hypergraph is an I-hypergraph. The proof of the result indicates an indirect connection between I-hypergraphs and interval graphs (i.e., intersection graphs of finite intervals on the real line), since both can be characterized in terms of an incidence matrix with the *consecutive ones* property.

Proposition 1 *Given a hypergraph H, one can decide in time proportional to*

$$|H| + SIZE(H)$$

whether or not H is (isomorphic to) an interval hypergraph.

Proof. Let H have h hyperedges. Consider the $|H| \times h$ (0,1)-valued *incidence matrix* for H, whose rows represent the vertices of H, whose columns represent the hyperedges of H, and whose (i, j)-th entry is 1 just when vertex i belongs to hyperedge j. One verifies easily that H is isomorphic to an interval hypergraph if, and only if, the rows of its incidence matrix can be permuted in such a way that all of the 1's in each column are consecutive. Booth and Lueker [5] present an algorithm that tests a (0,1)-valued matrix for this so-called *consecutive ones* property in time proportional to the sum of the number of rows (which is $|H|$) and the number of 1's (which is $SIZE(H)$), when the matrix is presented as a list of columns, with each column presented by a list of its 1-entries. □

3.2. Interval Graphs and Interval Hypergraphs

The proof of Proposition 1 has introduced the basic tool needed to expose the relationship between interval hypergraphs and their better-known relatives, interval graphs.

An *interval graph* is a graph G whose vertices can be put in one-to-one correspondence with intervals of the real line in such a way that vertices u and v are adjacent in G just when their corresponding intervals intersect.

Fulkerson and Gross [8] present a characterization of interval graphs in terms of a class of (0,1)-valued matrices. A *clique* in a graph G is a maximal set of mutually adjacent vertices. Given a graph G with c cliques, define the *clique vs. vertex incidence matrix* of G to be the $c \times |G|$ (0,1)-valued matrix whose rows represent the cliques of G, whose columns represent the vertices of G, and whose (i, j)-th entry is 1 just when vertex j belongs to clique i. Fulkerson and Gross establish the following.

Lemma 1 [8] *A graph is an interval graph if, and only if, its clique vs. vertex incidence matrix has the consecutive ones property.*

Thus, in some sense, the vertices of interval hypergraphs correspond to the cliques of interval graphs, while the hyperedges of interval hypergraphs correspond to the vertices of interval graphs. This correspondence can be made tight in one direction. Given a graph G, construct the hypergraph $H(G)$ as follows.

- For each clique κ of G, add a unique vertex $v(\kappa)$ to $H(G)$; every vertex of $H(G)$ arises in this way.
- For each vertex v of G, add a unique hyperedge to $H(G)$ containing every vertex $v(\kappa)$ of $H(G)$ for which the clique κ of G contains vertex v of G.

Proposition 2 *For any graph G, the clique vs. vertex incidence matrix of G is identical to the incidence matrix of the hypergraph $H(G)$. Hence, in particular, G is an interval graph if, and only if, $H(G)$ is an interval hypergraph.*

Thus, every interval hypergraph can be viewed as coming from an interval graph, and every interval graph spawns an interval hypergraph.

One can prove a weak converse of Proposition 2, showing how certain interval hypergraphs spawn interval graphs.

Given a hypergraph H, construct the graph $G(H)$ as follows.

- For each hyperedge η of H, there is a unique vertex $v(\eta)$ of $G(H)$; every vertex of $G(H)$ arises in this way.
- Two vertices $v(\eta_1)$ and $v(\eta_2)$ of $G(H)$ are adjacent just when the hyperedges η_1 and η_2 of H have nonempty intersection.

Say that the hypergraph H has the *vertex isolation property* if every vertex of H is the unique common element of some subset of the hyperedges of H.

Proposition 3 *For any hypergraph H having the vertex isolation property, the incidence matrix of H is identical to the clique vs. vertex incidence matrix of $G(H)$. Hence, in particular, $G(H)$ is an interval graph if, and only if, H is an interval hypergraph.*

The straightforward proofs of Propositions 2 and 3 are left to the reader.

3.3. Finding Small I-Hypergraphs

If the hypergraph H is not (isomorphic to) an I-hypergraph, one might wish to find the smallest I-hypergraph that contains H. Our next result indicates that finding this small I-hypergraph is likely to be computationally intractable, even when H is a graph. This demonstration exposes a connection between I-hypergraphs and the *Optimal Linear Arrangement Problem* for graphs.

Proposition 4 *The following problem is* NP*-complete: Given a graph G and an integer S, to decide whether or not there exists an I-hypergraph of SIZE S that contains G. The problem is solvable in polynomial time when G is a tree.*

Proof. The result will follow from a demonstration that our *Smallest Containing I-Hypergraph Problem* (**SCIH**) is *equivalent* to the Optimal Linear Arrangement Problem for graphs [10]

(OLA), in the sense that each problem is reducible to the other in polynomial time[4]. The reducibility of **OLA** to **SCIH** establishes the result for arbitrary graphs [10]; the reducibility of **SCIH** to **OLA** establishes the result for trees [18]. **OLA** is defined as follows.

OLA: Given a graph $G = (V, E)$ and an integer B, to decide whether or not there exists an injection $\lambda : V \rightarrow \{1, 2, \cdots, |G|\}$ for which

$$\sum_{(u,v)\in E} |\lambda(u) - \lambda(v)| \leq B.$$

The bases for our claimed reducibilities are the following correspondences. For any graph $G = (V, E)$ and injection $\lambda : V \rightarrow \{1, 2, \cdots, |G|\}$, G is embeddable, via the vertex-injection $\mu_v = \lambda$, in the I-hypergraph I_λ that has vertex-set $\{1, 2, \cdots, |G|\}$ and that has a hyperedge

$$\{\lambda(u), \lambda(u) + 1, \cdots, \lambda(v)\}$$

for each edge $(u, v) \in E$; in the embedding, μ_e maps the edge (u, v) to this hyperedge. The hypergraph I_λ is easily seen to have *SIZE*

$$|E| + \sum_{(u,v)\in E} |\lambda(u) - \lambda(v)|.$$

Conversely, let the graph $G = (V, E)$ be embedded in the I-hypergraph $I = (V_h, E_h)$ via the injections μ_v and μ_e. Then the injection $\lambda = \mu_v$ has

$$\sum_{(u,v)\in E} |\lambda(u) - \lambda(v)| \leq SIZE(I) - |E|,$$

since for each edge (u, v), $|\lambda(u) - \lambda(v)| \leq |\mu_e(u, v)| - 1$. The constructions outlined here can obviously be effected in polynomial time.

These correspondences demonstrate that the graph $G = (V, E)$ admits a linearization with **OLA**-cost B if, and only if, G is embeddable in an I-hypergraph of size $B + |E|$. It follows that the problems **OLA** and **SCIH** can each be efficiently reduced one to the other. □

3.4. Deciding Embeddability

Lacking the ability to determine efficiently how small an I-hypergraph the graph G can be embedded in, one might want at least to determine if G can be embedded in a given I-hypergraph I. We now show that this problem, too, is likely to be computationally intractable. This result illustrates a connection between I-hypergraphs and the *Bandwidth Minimization Problem* for graphs.

[4]Since all of the reductions we present in this section are "in polynomial time", we shall henceforth leave the phrase to be understood implicitly.

Proposition 5 *The following problem is* NP-*complete: Given a graph G and an I-hypergraph I, to decide whether or not there exists an embedding of G in I. The problem remains* NP-*complete even when G is a binary tree.*

Proof. The result will follow from a demonstration that the Bandwidth Minimization Problem for graphs (**BMP**) is reducible to our *I-Hypergraph Embeddability Problem* (**IHEP**). This reducibility establishes the result for arbitrary graphs because of [13], and for trees because of [9]. **BMP** is defined as follows.

BMP: Given a graph $G = (V, E)$ and an integer $B < |G|$,[5] to decide whether or not there exists an injection $\lambda : V \to \{1, 2, \cdots, |G|\}$ for which

$$\max_{(u,v)\in E} |\lambda(u) - \lambda(v)| \leq B.$$

The base for our claimed reducibility is the following correspondence. For any graph $G = (V, E)$ and integer B, construct the I-hypergraph $I_{G,B}$ that has vertex-set $\{1, 2, \cdots, |G|\}$ and that has, for each $1 \leq i \leq |G| - B + 1$, $\min\left(\binom{B}{2}, |E|\right)$ copies of the hyperedge

$$\{i, i+1, \cdots, i+B-1\}.$$

One verifies easily that the hypergraph $I_{G,B}$ can be constructed in time polynomial in the size of the description of G and B, since

$$SIZE(I_{G,B}) \leq |G|^2 \cdot \min\left(\binom{B}{2}, |E|\right) < |G|^4.$$

To establish the reduction, assume first that the graph G admits a layout λ for which

$$\max_{(u,v)\in E} |\lambda(u) - \lambda(v)| \leq B.$$

Then G is embeddable in $I_{G,B}$: The vertex-injection is $\mu_v = \lambda$; the edge-injection is defined by: $\mu_e(u, v)$ is an arbitrary one of the hyperedges of the form

$$\{\lambda(u), \lambda(u) + 1, \cdots, \lambda(u) + B - 1\}.$$

(We have endowed $I_{G,B}$ with ample copies of the hyperedge to guarantee that we can construct the injection μ_v.) Conversely, say that G is embeddable in $I_{G,B}$ via the injections μ_v and μ_e. Then the layout $\lambda = \mu_v$ has

$$|\lambda(u) - \lambda(v)| \leq |\mu_e(u, v)| = B$$

for each edge (u, v) of G, so G has bandwidth no greater than B. The result follows. □

[5]The inequality "$B < |G|$" is not usually included in the definition of **BMP**, but when it does not hold, the decision problem trivializes.

3.5. Deciding 2-Colorability

A hypergraph H is *2-colorable* if there is a way to assign one of two distinct colors to each vertex in H in such a way that no hyperedge of H remains monochromatic. Lovasz [12] has shown that the problem of deciding, given an arbitrary hypergraph H, whether or not H is 2-colorable is NP-complete. When H is a graph, the same question is easily decided in time $O(|V| + |E|)$, for a graph is 2-colorable if and only if it is bipartite. Deciding 2-colorability of I-hypergraphs is even easier, as the following observation verifies.

Proposition 6 *Every I-hypergraph is 2-colorable.*

Proof. Since every hyperedge of an I-hypergraph contains at least two vertices, and since the vertices of each hyperedge are "contiguous", one can 2-color an I-hypergraph by assigning one color to its odd-numbered vertices and another color to its even-numbered vertices. □

3.6. Deciding Connectivity and Path-Connectivity

An n-vertex hypergraph H is *connected* (resp., *path-connected*) if H contains an n-vertex tree (resp., the n-vertex path-graph P_n). The problem of deciding whether or not a given general hypergraph is path-connected is NP-complete, since it subsumes the problem of testing a graph for the existence of a hamiltonian path. In contrast, one can test a given I-hypergraph for path-connectivity via a straightforward efficient algorithm, because an I-hypergraph contains a spanning path-graph if, and only if, the path-graph is embeddable in the "natural" way. Moreover, again in contrast to arbitrary hypergraphs (or graphs, for that matter), an I-hypergraph is connected if, and only if, it is path-connected. We approach our decision algorithm via the following chain of lemmas.

Lemma 2 *Let the n-vertex tree T be embedded in the plane, with its vertices in a row, in the order*

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

Then: for each $1 \leq k < n$, at least k edges of T have one vertex in the set $\{v_1, v_2, \cdots, v_k\}$; at least one of these edges must also have a vertex in the set $\{v_{k+1}, v_{k+2}, \cdots, v_n\}$. Perforce, the same is true for the path-graph P_n, since it is an n-vertex tree.

Proof. Let us count the edges of T that contain[6] a vertex in the set $V_k =_{\text{def}} \{v_1, v_2, \cdots, v_k\}$. Say that V_k contains l connected components $C_1, C_2, \ldots, C_l$ of T, of sizes $s_1, s_2, \ldots, s_l$, respectively. By dint of its being connected, each component C_i must contain both vertices of at least $s_i - 1$ edges of T. Since T is connected, each of these components must also contain at least one vertex belonging to an edge whose other vertex resides in the set $\{v_{k+1}, v_{k+2}, \cdots, v_n\}$. The result now follows by counting. □

[6] This terminology is justified by our defining an edge as a two-element set.

Lemma 3 *Let the n-vertex I-hypergraph I contain an n-vertex tree T. Then, for each* $1 \leq k < n$, *at least k hyperedges of I have their smallest numbered vertex in the set* $\{1, 2, \cdots, k\}$; *at least one hyperedge has its highest numbered vertex in the set* $\{k+1, k+2, \cdots, n\}$. *Perforce, the same is true if I contains the path-graph* P_n, *since it is an n-vertex tree.*

Proof. Any embedding of T in I induces a layout of T in the plane with the vertices lying in a row; just let each $v_i = \mu_v^{-1}(i)$. Since each hyperedge of I realizes just one edge of T, the result is immediate from Lemma 2. □

Lemma 4 *If the n-vertex I-hypergraph I contains an n-vertex tree (perforce, if it contains the path-graph* P_n*), then I contains the path-graph* P_n *embedded via the vertex-injection* $\mu_v : V_{P_n} \rightarrow V_I$ *defined by*

$$\mu_v(i) = i$$

for $1 \leq i \leq n$.

Proof. It follows by an induction based on Lemma 3 that the following greedy algorithm specifies a valid edge-injection $\mu_e : Edges(P_n) \rightarrow Hyperedges(I)$ to complement the injection μ_v defined in the statement of the Lemma.

Proceed along the row of vertices from left to right. For each vertex i, assign edge $(i, i+1)$ to any hyperedge of I that

- contains vertices i and $i+1$
- has not yet been used
- has minimal largest element among hyperedges that have not yet been used

Details are left to the reader. □

Proposition 7 *Given an I-hypergraph I of SIZE* $s = SIZE(I)$, *one can determine in time* $O(s \cdot \log s)$ *whether or not I is connected or, equivalently, path-connected.*

Proof. Let I have h hyperedges. Say that I is presented via the $h \times 2$ matrix M_I that associates with each hyperedge of I its minimum and its maximum element: this matrix is clearly obtained from the incidence matrix of I in time linear in $SIZE(I)$. Reorder the columns of M_I so that the hyperedges of I are ordered by increasing minimum element and, among hyperedges with the same minimum element, by increasing maximum element. This reordering requires at most time $O(H(h))$, which is obviously $O(H(SIZE(I)))$. Once M_I is so rearranged, it is a simple matter to implement the algorithm of Lemma 4 in time linear in $SIZE(I)$. □

3.7. Finding a "Good" I-Hypergraph

Finally, we indicate how to construct a small I-hypergraph that contains a given graph G, based on a separator for G. This construction yields one more connection between the problem of embedding graphs in I-hypergraphs and the general problem of finding collinear layouts of graphs.

Proposition 8 *Let the graph G have an α-separator of size $S(n)$ for some $1/2 \leq \alpha < 1$. Then G is embeddable in an I-hypergraph $I(G)$ of SIZE at most*

$$|G| \cdot \left(\sum_{i=0}^{\lambda(|G|)} S(\alpha^i |G|) \right),$$

where $\lambda(|G|) = \log_{1/\alpha}(|G|)$.

Proof. We employ a strategy derived from [15]. Given a graph G, we construct $I(G)$ and the embedding-injections μ_v and μ_e as follows.

The I-hypergraph $I(G)$ has vertex-set $\{1, 2, \cdots, |G|\}$. To specify the injection μ_v, construct an $S(n)$-decomposition tree for G, as described in Section 2. Place the vertices of G in a row in the order they occur as leaves of the decomposition tree. This ordering implicitly specifies μ_v; it also implicitly lays out, in contiguous blocks, the vertices of all of the subgraphs of G that occur in the decomposition tree.

We now specify the hyperedges of $I(G)$ and the injection μ_e. For each of the (at most $S(|G|)$) edges that interconnect the two subgraphs G_1 and G_2 of G, at level-1 of the decomposition tree, give $H(G)$ a hyperedge $\{1, 2, \cdots, |G|\}$; let μ_e associate each connecting edge with a unique one of these hyperedges. Next: For each of the (at most $S(\alpha|G|)$) edges that interconnect the subgraphs G_{11} and G_{12} of G, at level-2 of the decomposition tree, give $I(G)$ the hyperedge $\{1, 2, \cdots, |G_1|\}$, and let μ_e associate each connecting edge with a unique one of these hyperedges; similarly, for each of the (at most $S(\alpha|G|)$) edges that interconnect the subgraphs G_{21} and G_{22} of G, at level-2 of the decomposition tree, give $I(G)$ the hyperedge $\{|G_1| + 1, |G_1| + 2, \cdots, |G|\}$, and let μ_e associate each connecting edge with a unique one of these hyperedges. We continue in the indicated fashion to add hyperedges to $H(G)$ for "routing" the interconnections among the subgraphs of G in the decomposition tree, using at most $S(\alpha^k|G|)$ copies of each hyperedge for the 2^{k-1} pairs of subgraphs at level-k of the tree. Once having completed this construction, we shall have constructed $I(G)$ and embedded G in it. It is clear from the construction that *SIZE*$(I(G))$ is bounded as claimed in the statement of the Proposition. □

4. STRONGLY UNIVERSAL I-HYPERGRAPHS

We turn now to the main result of this paper. Throughout this section, assume that we have been given the desired family of graphs Γ, where the largest graph in Γ has m vertices.

For convenience, say that $m = 2^r$ is a power of 2. Let Γ have an α-separator of size $S(n)$ for some $1/2 \leq \alpha < 1$.

Theorem 1 *Let the family of graphs Γ, as described above, be given. There is a strongly universal I-hypergraph $I(\Gamma)$ for Γ of size*

$$SIZE(I(\Gamma)) = m \cdot \left(\sum_{k=1}^{r} \sum_{i=0}^{-k/\log\alpha} S(\alpha^i 2^k) \right). \qquad (1)$$

The remainder of the section is devoted to proving Theorem 1, i.e., describing $I(\Gamma)$ and verifying that it is indeed strongly universal for Γ.

> **Remark.** In order to reconcile Equation (1) with the expression in the Abstract, recall that $-k/\log\alpha = \log_{1/\alpha} 2^k$.

4.1. The Construction of $I(\Gamma)$ and the Embedding Procedure

Let the vertices of $I(\Gamma)$ be the set $V_m = \{1, 2, \cdots, m\}$. We give $I(\Gamma)$ the following hyperedges: for $k = 1, 2, \cdots, r$ and $a = 0, 1, 2, \cdots, 2^{r-k} - 1$, we create $\sum_{i=0}^{-k/\log\alpha} S(\alpha^i 2^k)$ copies of the hyperedge

$$\{a2^k + 1, a2^k + 2, \cdots, (a+1)2^k\}.$$

It is clear that the $I(\Gamma)$ just constructed has size

$$SIZE(I(\Gamma)) = m \cdot \left(\sum_{k=1}^{r} \sum_{i=0}^{-k/\log\alpha} S(\alpha^i 2^k) \right),$$

as claimed in the Theorem.

Although formal validation of $I(\Gamma)$ will wait for the next subsection, we indicate informally how the graphs in Γ are embedded in arbitrary vertex-subsets of $I(\Gamma)$. Say that we are told that the p vertices

$$v_1, v_2, \cdots, v_p$$

(each $v_j \in \{1, 2, \cdots, m\}$) of $I(\Gamma)$ are the available ones and that we are to realize the ($\leq p$)-vertex graph $G \in \Gamma$ on these vertices. We begin the embedding process by constructing a $S(n)$-decomposition tree for G. We then lay out the vertices of G on the available vertices of $I(\Gamma)$, in the order in which the vertices occur as leaves of the $S(n)$-decomposition tree. (If G has fewer than p vertices, we arbitrarily choose $|G|$ of the available vertices for G's vertices.) Thus we have the vertex-injection μ_v. In order to specify the edge-injection μ_e, we associate with edge (u, v) of G any as-yet unused smallest hyperedge of $I(\Gamma)$ that contains both $\mu_v(u)$ and $\mu_v(v)$.

4.2. The Construction Validated

We now validate the construction and embedding process of the previous subsection. Our validation uses a new graph-theoretic notion motivated by the stringent demands of strong universality.

Strong Separation Profiles. Our interval hypergraphs $I(\Gamma)$ decompose naturally by bisection. Removing the largest hyperedges decomposes $I(\Gamma)$ into two copies of the I-hypergraph that we would construct if all graphs of size $> m/2$ were removed from Γ, and so on. When a graph G is embedded in $I(\Gamma)$, it is not clear how this bisection will dissect G, for that depends on which vertices of $I(\Gamma)$ are declared available for the embedding. Our guarantee that G can be embedded no matter which vertices of $I(\Gamma)$ are available thus leads naturally to the following notion.

Assume throughout that n (the number of vertices in $I(\Gamma)$) is a power of 2. Let G be a graph with n or fewer vertices, and let l be any integer $\geq \log n$. The l-tuple of nonnegative integers

$$\langle e_l, e_{l-1}, \cdots, e_1 \rangle$$

is a ***strong separation profile*** (an *SSP*, for short) for G, if the following property holds.

> *The SSP Property:* Given any integer n_1 such that both n_1 and $|G| - n_1$ are $\leq n/2$: By removing at most e_l edges from G, one can partition G into subgraphs G_1 having n_1 vertices and G_2 having $|G| - n_1$ vertices, each of which has $\langle e_{l-1}, e_{l-2}, \cdots, e_1 \rangle$ as an SSP. This recursive decomposition of G continues until we get down to subgraphs of G having at most one vertex.

Note that one can view each candidate decomposition of G (corresponding to the different choices for n_1) in terms of an $\langle e_l, e_{l-1}, \cdots, e_1 \rangle$*-decomposition tree* for G: the tree's root is G, with sons G_1 and G_2, and so on, just as with the $S(n)$-decomposition trees of the earlier sections.

The "strong" in the term SSP is intended to contrast with the notion of α-SP, wherein one seeks a "small cut" partition for just the case $\lfloor (1-\alpha)|G| \rfloor \leq n_1 \leq \lceil \alpha |G| \rceil$, rather than for all values of n_1, $1 \leq n_1 \leq n/2$.

The relevance of the notion of SSP resides in the following result.

Lemma 5 *Given any l-tuple of nonnegative integers*

$$\tau = \langle e_l, e_{l-1}, \cdots, e_1 \rangle$$

one can construct an $(m = 2^l)$-vertex I-hypergraph $I(m)$ of size

$$SIZE(I(m)) = m \cdot \sum_{i=1}^{l} e_i$$

that is strongly universal for the family $\Gamma(\tau)$, where $\Gamma(\tau)$ comprises all graphs having the tuple τ as an SSP.

Proof.

The I-Hypergraph $I(m)$

To construct $I(m)$, we create the following hyperedges from the vertex-set $V_m = \{1, 2, \cdots, m\}$. For $k = 1, \cdots, l$ and $a = 0, 1, 2, \cdots, 2^{l-k} - 1$, we create e_k copies of the hyperedge

$$\{a2^k + 1, a2^k + 2, \cdots, (a+1)2^k\}.$$

It is clear that $I(m)$, so constructed, has the claimed *SIZE*.

The Embedding Procedure

Say that we are told that the p vertices

$$v_1, v_2, \cdots, v_p$$

(each $v_j \in \{1, 2, \cdots, m\}$) of $I(m)$ are available and that we are to embed the ($\leq p$)-vertex graph $G \in \Gamma(\tau)$ on these vertices. The essence of the embedding process is the construction of an $\langle e_l, e_{l-1}, \cdots, e_1 \rangle$-decomposition tree for G. We begin by choosing some $|G|$ of the available vertices of $I(m)$ upon which to place the vertices of G; these vertices can be chosen ***in any way whatsoever***. This choice then determines the parameter n_1, which is the size of one of the two graphs we shall partition G into: Specifically,

$$n_1 =_{\text{def}} |\{v_j : v_j \leq 2^{l-1}\}|;$$

i.e., n_1 is the number of selected available vertices that reside to the left of the midpoint $m/2$ of $I(m)$. By definition of SSP, G can be partitioned into a subgraph of size n_1 and one of size $|G| - n_1$ by removing no more than e_l edges from G. These edges can thus be embedded in the e_l size-m hyperedges of $I(m)$, no matter which vertices of $I(m)$ their endpoints are placed on. By definition of SSP, we may assume that each of the two resulting subgraphs has an SSP

$$\langle e_{l-1}, e_{l-2}, \cdots, e_1 \rangle.$$

We thus find ourselves with two half-size versions of our original problem: By removing the e_l large hyperedges from $I(m)$, we are left with two copies of $I(m/2)$ in which to embed the two subgraphs of G, each by definition having no more than 2^{l-1} vertices. We leave to the reader the easy details of inductively validating this recursive embedding process (which can be viewed as building an $\langle e_l, e_{l-1}, \cdots, e_1 \rangle$-decomposition tree for G). □

Determining SSPs for arbitrary graphs is not a trivial pursuit. However, one can, with little difficulty, discover profiles for certain familiar graphs. For instance, every ($\leq n$)-vertex binary tree has an SSP of the form

$$\langle \log n, \log(n/2), \cdots, 1 \rangle$$

so $e_{k-1} = e_k - 1$; similarly, every ($\leq n$)-vertex rectangular mesh has an SSP of the form

$$\langle \sqrt{n}, \sqrt{n}/\sqrt{2}, \sqrt{n}/2, \cdots, 1 \rangle$$

so $e_{k-1} = e_k/\sqrt{2}$.[7] The following Lemma helps one discover SSPs; and it combines with Lemma 5 to complete the proof of Theorem 1.

Lemma 6 *Let Γ be a family of graphs having an α-separator of size $S(n)$. For every integer r, every graph $G \in \Gamma$ with $|G| \leq 2^r$ has an SSP*

$$\langle e_r, e_{r-1}, \cdots, e_1 \rangle,$$

where each

$$e_k = \sum_{i=0}^{-k/\log \alpha} S(\alpha^i 2^k).$$

Proof. The proof builds on the technique used in the proof of Proposition 8 for laying out (within the groundrules of Section 2.2) any given $G \in \Gamma$; therefore, we shall be very sketchy here. Note that the layout here (in contrast to that in Proposition 8) is purely a technical device and should not be construed as an embedding of G in an I-hypergraph, despite the formal similarity.

Construct an $S(n)$-decomposition tree for G, and place the vertices of G in a row in the order they occur as leaves of the decomposition tree. Run $S(|G|)$ routing tracks above the vertices, in which to route the edges that interconnect the two subgraphs G_1 and G_2 of G at level-1 of the decomposition tree. These routing tracks can be viewed as rows in the plane that are reserved for "drawing" the edges of G; thus every edge of G ends up being drawn as two vertical line segments from its terminal vertices to the associated routing track, plus a horizontal line segment (in the routing track) joining the two vertical segments. Then run $S(\alpha|G|)$ routing tracks over the vertices of G_1 and the same number of routing tracks over the vertices of G_2. Continue in the indicated fashion to run routing tracks for routing the edges among the subgraphs of G in the decomposition tree, using $S(\alpha^k|G|)$ routing tracks for the 2^{k-1} pairs of subgraphs at level-k of the tree. The reader will note that we have constructed here a layout of G that uniformly has

$$W = \sum_{i=0}^{\log_{1/\alpha}|G|} S(\alpha^i|G|)$$

routing tracks above every vertex. It follows that, given any integer $n \leq |G|$, G can be partitioned into a subgraph of size n and one of size $|G| - n$ by removing (or "cutting") at most W edges. In particular, such a partition is possible for any n such that both n and $|G| - n$ are $\leq 2^{r-1}$. □

Lemmas 5 and 6 combine to establish Theorem 1.

[7]The cited SSPs for trees and meshes can be derived by considering the sizes of "perimeters" of regions within the graphs.

4.3. The Issue of Optimality

There are many families of graphs for which our strongly universal I-hypergraphs are within a constant factor of optimal in *SIZE*. We cite two major examples.

Let us restrict attention to ***honest*** separation functions for families Γ, i.e., separator functions $S(n)$ that truly reflect the difficulty of cutting the member graphs into pieces, in the sense that $\Theta(S(n))$ edges are necessary, as well as sufficient, to partition an n-vertex graph in Γ into two subgraphs of appropriate sizes.

Binary Trees. It is shown in [6] that any I-hypergraph that is strongly universal for the family of binary trees (which admits the honest (2/3)-separator function $S(n) \equiv 1$) has *SIZE* $\Omega(n \cdot \log^2 n)$, which is within a constant factor of the *SIZE* of the I-hypergraph produced by the construction in the proof of Theorem 1.

Algebraic Separators. Let the family Γ have an honest α-separator of size $S(n) = n^\delta$ for some constant δ. The double summation (1) in Theorem 1 becomes a double geometric sum, so the construction in the proof of the Theorem yields an I-hypergraph of *SIZE* $O(n^{1+\delta})$. On the other hand, invoking the honesty of $S(n)$, we can invoke the bounding techniques of [15] to show that any I-hypergraph that is strongly universal for Γ must have *SIZE* $\Omega(n^{1+\delta})$; indeed any collinear layout of the graphs in Γ must occupy this much area, so in this case, there is at most constant factor overhead for the fault tolerance afforded by the strong universality of the I-hypergraphs we produce.

4.4. A Remaining Challenge

In [17], we studied the strong universality problem for the family Π_n of path-graphs containing at most n vertices. We were able to show there that one could sometimes produce "strongly universal" I-hypergraphs of smaller *SIZE*, if one moderated one's demands on strong universality so that one was guaranteed to be able to embed a given G from the target graph family Γ (Π_n in that paper) ***only with very high probability***. Specifically, the following results appear in that paper.

Proposition 9 [17]
(a) *For all n, there exists an n-vertex I-hypergraph I_n of SIZE $O(n \cdot \log n)$ that is strongly universal for the family Π_n.*
(b) *Any n-vertex I-hypergraph that is strongly universal for the family Π_n must have SIZE $\Omega(n \cdot \log n)$.*

Now, let us change the game somewhat by assuming that when we "kill" vertices of the I-hypergraphs in question (i.e., make them unavailable), we do so independently, with probability 1/2. We can now consider the situation where an I-hypergraph is strongly universal with some given probability (which depends on the probabilities of certain patterns of "kills"). The following result concerns such a scenario.

Proposition 10 [17]
(a) *For all n, there exists an n-vertex I-hypergraph J_n of SIZE $O(n \cdot \log\log n)$ that is strongly universal for the family Π_n, with probability*

$$1 - \frac{1}{2n \cdot \log n}.$$

(b) *For $n > 4$, any n-vertex I-hypergraph that is strongly universal for the family Π_n with probability at least*

$$1 - \frac{1}{2n \cdot \log n}$$

must have SIZE $\Omega(n \cdot \log\log n)$.

In order to lend intuition to the reader, we sketch the proofs of these results very briefly.

The nonprobabilistic upper bound of Proposition 9(a) proceeds much as in the proof of Theorem 1 in Section 4.1: Assume for simplicity that n is a power of 2. For $k = 1, \cdots, l$ and $a = 0, 1, 2, \cdots, 2^{l-k} - 1$, we endow I_n with one copy of the hyperedge

$$\{a2^k + 1, a2^k + 2, \cdots, (a+1)2^k\}.$$

The proof that I_n is strongly universal for Π_n consists of showing that one can always embed any sufficiently small path-graph by associating vertices of the path-graph with available vertices of I_n in any order-preserving way, and realizing each edge of the path-graph via the smallest hyperedge that contains the images under μ_v of the edge's endpoints.

The nonprobabilistic lower bound of Proposition 9(b) proceeds by noting that, given any n-vertex I-hypergraph I that is strongly universal for the family Π_n, we must be able to embed the 2^{k+1}-vertex path-graph in I, using vertices

$$1, n/2^k, n/2^k + 1, \cdots, n - n/2^k, n - n/2^k + 1, n$$

of I, for each $k \in \{0, 1, \cdots, \log n\}$. Each value of k thus contributes roughly n to *SIZE*(I), for a total of $\Omega(n \cdot \log n)$. (One must take some care in counting these contributions to *SIZE*(I), since hyperedges added to satisfy the requirements of a small value of k can be reused in satisfying the requirements of larger values of k.)

The probabilistic upper bound of Proposition 10(a) follows from a composite construction of J_n. Assume as before that n is a power of 2. Partition the set $\{1, 2, \cdots, n\}$ into contiguous blocks of length $m = 2\log n$ each. Assign hyperedges to each block to make it a copy of I_m, as described above. Additionally, for each pair of adjacent blocks, add a hyperedge that is the union of the vertices in the two blocks. One now verifies that given any selection of available vertices of J_n, one can embed a path-graph using all of those vertices *unless* there are two available vertices separated by a block containing no available vertex. However, the probability of such an occurrence is no greater than $1/(2n \cdot \log n)$.

Finally, to see the probabilistic lower bound of Proposition 10(b), partition the vertices of any given n-vertex I-hypergraph I into contiguous blocks of $m = \log n + \log\log n + 1$ vertices each. Assume that for each block B, there is some pattern of "killed" vertices that makes it impossible to embed a path-graph on all of the available vertices. The probability that I can embed any sufficiently small path-graph cannot exceed the probability that one of these bad patterns occurs, which the reader can easily show to be greater than $1/(2n \cdot \log n)$. As a consequence, one can show that if I successfully embeds path graphs with probability exceeding $1 - 1/(2n \cdot \log n)$, then it must *always* work on every one of the blocks B, hence must have $SIZE$ at least as great as I_m (by Proposition 9(b)).

Details on all four proofs are found in [17].

The Challenge: We are certain that savings analogous to those exposed in Propositions 9 and 10 are attainable with strongly universal I-hypergraphs for a large variety of graph families other than Π_n, but we have as yet been unable to generalize this phenomenon even to binary trees. Such generalization is an inviting challenge.

ACKNOWLEDGMENT. It is a pleasure to thank Lenny Heath and Bruce Leban for a careful reading of the manuscript.

5. REFERENCES

1. N. Alon and F.R.K. Chung (1985): Explicit constructions of linear-sized fault-tolerant networks. Typescript, MIT.

2. J. Beck (1983): On size Ramsey number of paths, trees, and circuits, I. *J. Graph Th.* *7*, 115-129.

3. J. Beck (1983): On size Ramsey number of paths, trees, and circuits, II. Manuscript.

4. S.N. Bhatt and C.E. Leiserson (1984): How to assemble tree machines. In *Advances in Computing Research 2*, (F.P. Preparata, ed.) JAI Press, Greenwich, CT, pp.95-114.

5. K.S. Booth and G.S. Lueker (1976): Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *J. Comput. and Syst. Sci.* *13*, 335-379.

6. F.R.K. Chung and A.L. Rosenberg (1986): Minced trees, with applications to fault-tolerant VLSI processor arrays. *Math. Syst. Th. 19*, 1-12.

7. J. Friedman and N. Pippenger (1986): Expanding graphs contain all small trees. Typescript, IBM Almaden Research Center.

8. D.R. Fulkerson and O.A. Gross (1965): Incidence matrices and interval graphs. *Pacific J. Math. 15,* 835-855.

9. M.R. Garey, R.L. Graham, D.S. Johnson, D.E. Knuth (1978): Complexity results for bandwidth minimization. *SIAM J. Appl. Math. 34*, 477-495.

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

11. R.J. Lipton and R.E. Tarjan (1979): A separator theorem for planar graphs. *SIAM J. Appl. Math. 36*, 177-189.

12. L. Lovasz (1973): Coverings and colorings of hypergraphs. *4th Southeast Conf. on Combinatorics, Graph Theory, and Computing.* Utilitas Mathematica Publ., Winnipeg, 3-12.

13. C.H. Papadimitriou (1976): The *NP*-completeness of the bandwidth minimization problem. *Computing 16*, 263-270.

14. G.L. Peterson and Y.-H. Ting (1982): Trade-offs in VLSI for bus communication networks. Tech. Rpt. 111, Univ. Rochester.

15. A.L. Rosenberg (1981): Routing with permuters: Toward reconfigurable and fault-tolerant networks. Tech. Rpt. CS-1981-13, Duke Univ.

16. A.L. Rosenberg (1984): On designing fault-tolerant VLSI processor arrays. In *Advances in Computing Research 2*, (F.P. Preparata, ed.) JAI Press, Greenwich, CT, pp.181-204.

17. A.L. Rosenberg (1985): A hypergraph model for fault-tolerant VLSI processor arrays. *IEEE Trans. Comp., C-34*, 578-584.

18. Y. Shiloach (1979): A minimum linear arrangement algorithm for undirected trees. *SIAM J. Comput. 8*, 15-32.

19. Q. Stout (1986): Meshes with multiple busses. *27th IEEE Symp. on Foundations of Computer Science*, 264-273.

20. L.G. Valiant (1981): Universality considerations in VLSI circuits. *IEEE Trans. Comp., C-30*, 135-140.