Embedding hyperpyramids into hypercubes

by C.-T. Ho
S. L. Johnsson

A $\hat{P}(k, d)$ hyperpyramid is a level structure of k hypercubes, where the hypercube at level i is of dimension id, and a node at level $i - 1$ is connected to every node in a d-dimensional subcube at level i, except for the leaf level k. Hyperpyramids contain pyramids as proper subgraphs. We show that a hyperpyramid $\hat{P}(k, d)$ can be embedded in a hypercube with minimal expansion and dilation = d. The congestion is bounded from above by $\lceil (2^d - 1)/d \rceil$ and from below by $1 + \lceil (2^d - d)/(kd + 1) \rceil$. We also present embeddings of a hyperpyramid $\hat{P}(k, d)$ together with $2^d - 2$ hyperpyramids $\hat{P}(k - 1, d)$ such that only one hypercube node is unused. The dilation of the embedding is $d + 1$, with a congestion of $O(2^d)$. A corollary is that a complete n-ary tree can be embedded in a hypercube with dilation = $\max(2, \lceil \log_2 n \rceil)$ and expansion = $(2^{k\lceil \log_2 n \rceil + 1})(n - 1)/(n^{k+1} - 1)$.

Introduction

Processor utilization and communication time are two important considerations in selecting data structures and algorithms for computer systems assembled from a large number of parts. Communication is one of the most expensive resources to be considered in such a system, and its efficient utilization is imperative. In studying the efficient utilization of the communication system, one can model the communication needs of the computations with a graph, which is referred to as the *guest graph* [1]. This graph describes the interaction between the data elements of the computation, where a node represents a process and an edge represents a communication need between the two connected processes. Similarly, the topology of the computer system is captured by the *host graph*. Each node represents a processor with local storage, and each edge represents a communication link between processors. For the purpose of planning the execution of a computation represented by guest graph G on a host represented by host graph H, an embedding function f is used to embed G into H. These graphs, as well as other items discussed in this section, are defined formally in later sections.

The embedding function f maps each node in the guest graph G into a unique node in the host graph H, and each edge in G into a path in H. Let $V(X)$ and $E(X)$ respectively denote the node set and the edge set of a graph X. Let $|S|$ denote the cardinality of a set S. The *expansion* of the mapping f is defined as $|V(H)|/|V(G)|$. It is a measure of processor utilization. The *dilation* of the mapping is defined as the maximum length of path $f(e_G)$ for all $e_G \in E(G)$, where e_G is mapped into the path $f(e_G)$ in H. The *congestion* of the mapping is defined as the maximum number of guest-graph edges sharing an edge in the host graph. The slowdown of nearest-neighbor communication in the guest graph caused by edges being "stretched" into paths of length greater than 1 is generally a function of the dilation and the congestion. Thus, the general goal of graph embeddings is, given a guest graph G and a host graph H, to find an embedding function f that minimizes

the dilation and congestion. In this paper, we discuss embedding of pyramids and hyperpyramids, to be defined later, into hypercubes, with minimal expansion and improved dilation and/or congestion over previous results. *Minimal expansion* means that the hypercube host graph is the smallest one that has as many nodes as the given guest graph.

Related to the embedding of pyramids is the embedding of meshes and trees. Embedding of meshes into hypercubes has been studied in [2–7]. Embedding of trees into hypercubes has been studied in [3, 8–18]. Several parallel algorithms that naturally lend themselves to a pyramid topology are discussed, for instance, in [19–23]. Multigrid algorithms for partial differential equations [24] and certain algorithms for image processing [22] are specific examples. The embedding of pyramids into hypercubes was first studied by Stout [25]. He proved that there exists an embedding with dilation = 1 of an M-node pyramid into an N-node hypercube with $N \ll M$, if $\approx M/N$ pyramid nodes are mapped into every hypercube node. Stout also showed that for a one-to-one mapping from a pyramid to a hypercube, minimal expansion and dilation = 2 is possible. Lai and White [26] gave embedding algorithms with dilation = 2 and congestion = 3, or dilation = 3 and congestion = 2 (both with minimal expansion). We give an embedding with dilation = 2, congestion = 2, and minimal expansion. We also generalize such an embedding to embeddings of hyperpyramids into hypercubes with minimal expansion and with dilation = d. Hyperpyramids of order d are graphs in which each nonleaf node has 2^d children, and the nodes at the same level form a hypercube (instead of a mesh).

Lai and White [27] also gave an algorithm for embedding a pyramid and two smaller pyramids (each with approximately a quarter of the size of the larger pyramid) into a hypercube, with expansion ≈ 1, dilation = 3, and congestion = 6. We improve the result to expansion ≈ 1, dilation = 3, and congestion = 3. The result is generalized to the embedding of one hyperpyramid with minimal expansion, and the embedding of $2^d - 2$ smaller hyperpyramids into the same hypercube, with a total expansion ≈ 1 and a dilation of $d + 1$.

Note that the technique used in our embeddings is quite different from that of Lai and White for both single and multiple (hyper)pyramid embeddings. Furthermore, a recent work by Ziavras et al. [28, 29], who implemented on a CM-2™ parallel system various known embeddings of pyramids into hypercubes, including ours and that of Lai and White, observed that a small improvement (such as from 3 to 2) in congestion or dilation sometimes implies significant improvement in performance.

In the next section, we introduce the notation used in the paper, define pyramids and hyperpyramids, and give some of their properties. The section on embedding hyperpyramids into hypercubes contains the main results, and the final section summarizes the paper.

Preliminaries

Let 0^m denote a string of m 0-bits, and let 1^m denote a string of m 1-bits. Let j_m be the mth bit of the binary representation of j, with the least significant bit being the 0th bit. Let $x^{\{m\}} = x \oplus 10^m$, i.e., x with the mth bit complemented. We use $(x|y)$ to denote the concatenation of two strings x and y. In the next two subsections, we define a few metrics used for graph embedding, and define a few relevant graphs and their related properties.

- *Graph embeddings*

Definition 1 An embedding f of a *guest* graph, G, into a *host* graph, H, is a one-to-one mapping from $V(G)$ to $V(H)$, combined with a mapping of the edges of $E(G)$ into simple paths in $E(H)$ so that if $e_G = (i, j) \in E(G)$, then $f(e_G)$ is a simple path in H with endpoints $f(i)$ and $f(j)$. The *expansion* of the embedding f is

$$exp_f = \frac{|V(H)|}{|V(G)|}.$$

Let $E[f(e_G)]$ denote the set of edges in the path $f(e_G)$.

Definition 2 The dilation of an edge $e_G \in E(G)$ is the length of the path $f(e_G)$:

$$dil_f(e_G) = |E[f(e_G)]|.$$

The *dilation* of the embedding f is

$$dil_f(G) = \max_{\forall e_G \in E(G)} dil_f(e_G).$$

We sometimes also consider dilation of a set of edges S as

$$dil_f(S) = \max_{\forall e_G \in S} dil_f(e_G).$$

Definition 3 The *congestion* of an edge $e_H \in E(H)$, $cong_f(e_H)$, is the number of edges in G mapped to paths that include e_H; i.e.,

$$cong_f(e_H) = \sum_{\forall e_G \in E(G)} |\{e_H\} \cap E[f(e_G)]|.$$

The *congestion* of the mapping f is

$$cong_f = \max_{\forall e_H \in E(H)} cong_f(e_H).$$

- *Graph definitions*

A mesh is a rectangular array of nodes, with edges connecting adjacent nodes.

Definition 4 An $l_1 \times l_2$ mesh $M(l_1, l_2)$ is a graph with node set

$$V[M(l_1, l_2)] = \{(x_1, x_2) | 0 \le x_1 < l_1, 0 \le x_2 < l_2\}$$

and edge set

$$E[M(l_1, l_2)] = \{(x, x') | x = (x_1, x_2), x' = (x_1', x_2') \in V[M(l_1, l_2)], |x_1 - x_1'| + |x_2 - x_2'| = 1\}.$$

Definition 5 A k-level pyramid $P(k, l_1, l_2)$ is a graph with node set

$$V[P(k, l_1, l_2)] = \bigcup_{i=0}^{k} \{(i, x_1, x_2) | (x_1, x_2) \in V[M(l_1^i, l_2^i)]\}$$

and edge set

$$E[P(k, l_1, l_2)] = \bigcup_{i=0}^{k} \{[(i, x_1, x_2), (i, x_1', x_2')] | [(x_1, x_2), (x_1', x_2')] \in E[M(l_1^i, l_2^i)]\} \cup \bigcup_{i=1}^{k} \left\{\left[(i, x_1, x_2), \left(i - 1, \left\lfloor \frac{x_1}{l_1} \right\rfloor, \left\lfloor \frac{x_2}{l_2} \right\rfloor\right)\right] | (x_1, x_2) \in V[M(l_1^i, l_2^i)]\right\}.$$

Intuitively, a $P(k, l_1, l_2)$ pyramid is made up of the graphs $M(l_1^0, l_2^0)$ through $M(l_1^k, l_2^k)$, with each node having $l_1 \times l_2$ children, except nodes at level k. Node $(i, x_1, x_2) \in V[P(k, l_1, l_2)]$ is at *level* i. The node at level 0, (0, 0, 0), is called the *apex*, or the *root* of the pyramid. The nodes at level k are leaf nodes, and the mesh at level k, $M(l_1^k, l_2^k)$, is the *base* of the pyramid. Clearly, $P(k, l_1, l_2)$ is isomorphic to $P(k, l_2, l_1)$. **Figure 1** shows the topology of the pyramid $P(2, 2, 2)$. It can be viewed as a complete quad-tree with nodes at the same level being connected as a mesh.

The number of nodes in a pyramid is

$$|V[P(k, l_1, l_2)]| = \sum_{i=0}^{k} (l_1 l_2)^i = \frac{(l_1 l_2)^{k+1} - 1}{l_1 l_2 - 1},$$

and the number of edges is

$$|E[P(k, l_1, l_2)]| = \sum_{i=1}^{k} [3(l_1 l_2)^i - l_1^i - l_2^i].$$

Definition 6 A d-dimensional hypercube, denoted Q_d, has 2^d nodes. Each node can be assigned a binary string of length d as a unique address, such that any two nodes are connected through an edge if and only if their addresses differ in exactly one bit.

Figure 2 shows the hypercube Q_4. The addresses of hypercube nodes in subsequent figures are omitted, but they are determined in the same way. For clarity, we also omit edges of high dimensions in subsequent figures.

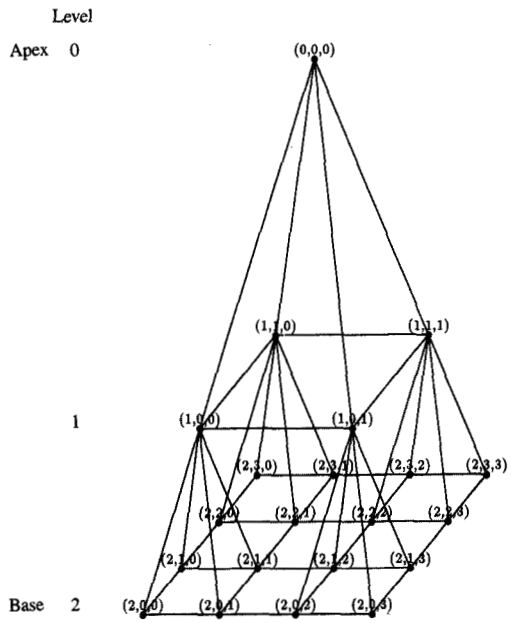

Figure 1

Topology of the pyramid $P(2, 2, 2)$.

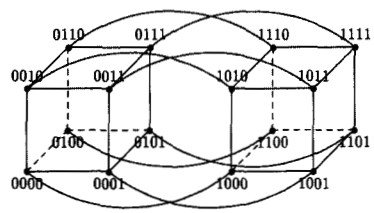

Figure 2

Hypercube Q_4, with node-addressing scheme used in this paper.

Definition 7 The *k-level hyperpyramid* of *degree d*, denoted $\hat{P}(k, d)$, is defined recursively as follows. $\hat{P}(0, d)$ is the root node. The hyperpyramid $\hat{P}(k, d)$ is constructed from 2^d hyperpyramids $\hat{P}(k - 1, d)$ by first

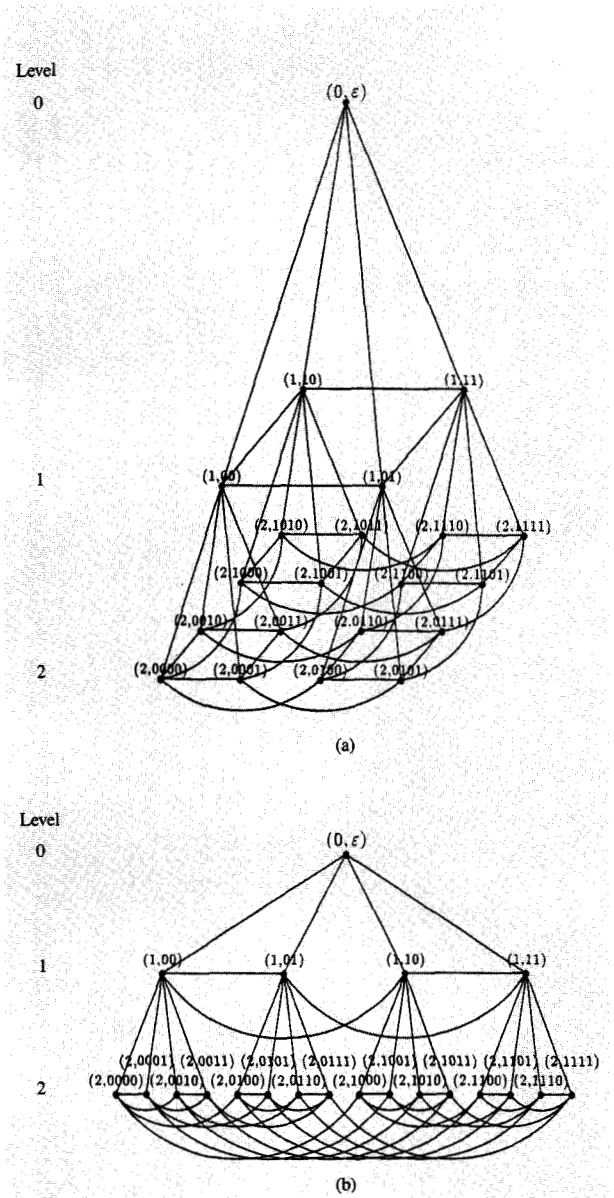

Figure 3

Two different views of the topology of the hyperpyramid $\hat{P}(2, 2)$.

interconnecting corresponding nodes in each of these hyperpyramids as d-dimensional hypercubes, and then creating a new root node and connecting it to every root of the hyperpyramids $\hat{P}(k - 1, d)$.

The nodes of hyperpyramid $\hat{P}(k, d)$ are given addresses (i, j) such that i identifies the level ($0 \le i \le k$) and j identifies one of the 2^{id} nodes at that level. Here, j is a binary number of length id. If $i \neq 0$, node (i, j) connects to a parent node $(i - 1, j_{id-1}\, j_{id-2} \cdots j_d)$, and, if $i \neq k$, to 2^d children nodes with addresses $\{(i + 1, j|*_{d-1} *_{d-2} \cdots *_0)\}$, where $*_m = 0$ or 1 for all $0 \le m < d$. (Recall that "|" is the concatenation operator.) The second argument of the parent address $(j_{id-1}\, j_{id-2} \cdots j_d)$ is obtained by removing the d lowest-order bits from j. The second arguments of the child addresses $(j|*_{d-1} *_{d-2} \cdots *_0)$ are obtained by appending d-bit binary strings to j. These edges form the "tree edges" of the hyperpyramid. In addition there are id "cube edges" connecting node (i, j) to nodes $(i, j^{\{m\}})$ for all $0 \le m < id$. (Recall that $j^{\{m\}}$ is j with the mth bit complemented.)

Figure 3(a) shows the topology of the hyperpyramid $\hat{P}(2, 2)$. Note that id bits are used for the second arguments of the node addresses at level i. The second argument of the root node is a null string, which is represented by ε. **Figure 3(b)** gives another view of the same hyperpyramid.

We use the hyperpyramid as an intermediate graph in embedding a pyramid into a hypercube. For the purpose of embedding, it is more convenient to assume that each mesh plane of a pyramid is a hypercube. Note that introducing the intermediate graph does not increase the dilation of our embedding. Furthermore, for certain multilevel algorithms on a domain of three dimensions or higher, the guest graph can be characterized by a hyperpyramid but not by a pyramid. Thus, hyperpyramid embeddings give this flexibility.

Proposition 1 A hyperpyramid $\hat{P}(k, d)$ contains a pyramid $P(k, 2^j, 2^{d-j})$, for all $0 \le j \le d$, as a subgraph.

Proof We define a one-to-one mapping from the node set of $P(k, 2^j, 2^{d-j})$ to the node set of $\hat{P}(k, d)$ as follows. Each node (i, x_1, x_2) in $P(k, 2^j, 2^{d-j})$, where $0 \le i \le k$, $0 \le x_1 < 2^j$, and $0 \le x_2 < 2^{d-j}$, is mapped to a unique node $[i, g_j(x_1)|g_{d-j}(x_2)]$ in $\hat{P}(k, d)$, where $g_j(x)$ is the binary-reflected Gray code of x in j bits. It is straightforward to verify that any two neighboring nodes in $P(k, 2^j, 2^{d-j})$ are mapped to adjacent nodes in $\hat{P}(k, d)$. ■

A corollary is that a hyperpyramid $\hat{P}(k, 2)$ contains a pyramid $P(k, 2, 2)$ as a subgraph. In the following we consider only the embedding of hyperpyramids into hypercubes.

We use Definition 7 in specifying embedding functions, f, and proving their properties with respect to dilation and congestion. Hyperpyramids can also be defined recursively by adding a hypercube Q_{kd} to a hyperpyramid $\hat{P}(k - 1, d)$. The hyperpyramid $\hat{P}(k, d)$ is obtained by connecting each node in Q_{kd} to a (parent) node in the base of the hyperpyramid $\hat{P}(k - 1, d)$. Such a definition emphasizes the fact that hyperpyramids can be viewed as a sequence of hypercubes of linearly increasing dimensions, with a tree structure connecting them.

The number of nodes in a hyperpyramid $\hat{P}(k, d)$ is

$$|V[\hat{P}(k, d)]| = \sum_{i=0}^{k} 2^{id} = \frac{2^{(k+1)d} - 1}{2^d - 1},$$

and the number of edges is

$$|E[\hat{P}(k, d)]| = \sum_{i=1}^{k} id2^{id-1} + \sum_{i=1}^{k} 2^{id}.$$

In the formula for the number of edges, the first term accounts for the edges at the levels and the second term accounts for the edges between the levels. From Figures 1 and 3(a), it is clear that a pyramid $P(2, 2, 2)$ with wraparound edges added to the mesh at level 2 is topologically equivalent to a hyperpyramid $\hat{P}(2, 2)$. This is because a 4×4 torus is topologically equivalent to the hypercube Q_4 (and, in general, a d-dimensional torus of form $4 \times 4 \times \cdots \times 4$ is topologically equivalent to the hypercube Q_{2d}).

Embedding hyperpyramids into hypercubes

The main results of this paper are the following:

1. A hyperpyramid $\hat{P}(k, d)$, with $d \geq 2$, can be embedded into the hypercube Q_{kd+1}, with expansion <2 and dilation $= d$. The congestion is bounded from below by $1 + \lceil (2^d - d)/(kd + 1) \rceil$ and from above by $\lceil (2^d - 1)/d \rceil$.
2. A hyperpyramid $\hat{P}(k, d)$ together with $(2^d - 2)$ hyperpyramids $\hat{P}(k - 1, d)$, $d \geq 2$, can be embedded into a hypercube Q_{kd+1} with expansion ≈ 1 (only one hypercube node is not used) and dilation $= d + 1$. The congestion is at most $O(2^d)$.

These two embeddings are described in the next two subsections. For the purpose of defining embeddings based on induction, we use a two-rooted hyperpyramid defined next.

Definition 8 A *two-rooted hyperpyramid* $\bar{P}(k, d)$ is a hyperpyramid $\hat{P}(k, d)$ with an additional root node and additional edges between the additional node and all nodes at level 1. The two roots are denoted $(0, \varepsilon)$ and $(0', \varepsilon)$, respectively.

Since the two roots are symmetrical, either one can serve as the root of the hyperpyramid. One of the two roots will be a node at level 1 after the induction step. This root is called the *real root*. The other root will either serve as one of the two *new* roots or become unused after the induction step. This root is called the *spare root*. There is no edge between the two roots according to Definition 8, but the embedding functions presented below always map the two roots to adjacent hypercube nodes. The idea of using two roots for the recursive construction of tree structures has been used before by Bhatt and Leiserson [30], for instance, in constructing a complete binary tree out of "chips" containing smaller trees, and by Bhatt and Ipsen [12] in embedding a complete binary tree into a hypercube.

• *Embedding a hyperpyramid into a hypercube*

In this subsection, we give an embedding of $\hat{P}(k, d)$ into Q_{kd+1} with dilation $= d$ and congestion $= \lceil (2^d - 1)/d \rceil$. We also show some lower bounds in dilation and congestion over all possible embeddings. We define the embedding by induction and prove the upper bounds on dilation and congestion of our embedding, also by induction. Although the upper bound of dilation itself can be derived using a much simpler proof, such as one based on Equation (1), which is given later, the inductive step in the next theorem is required for proving the bound on congestion.

Dilation

Theorem 1 A hyperpyramid $\hat{P}(k, d)$, with $d \geq 2$, can be embedded into a hypercube Q_{kd+1} with dilation $= d$.

Proof Instead of considering the embedding of a hyperpyramid $\hat{P}(k, d)$, we consider the embedding of the corresponding two-rooted hyperpyramid $\bar{P}(k, d)$. The dilation for the two-rooted hyperpyramid is an upper bound on the dilation for the corresponding hyperpyramid with a single root, as the latter is a subgraph of the former. We define a function f_k, which maps a two-rooted hyperpyramid $\bar{P}(k, d)$ into hypercube Q_{kd+1}, with dilation $= d$, by a recursive construction on k and prove the theorem by induction. The induction hypothesis is that for $k \leq n$, a two-rooted hyperpyramid $\bar{P}(k, d)$ can be embedded by f_k into a hypercube Q_{kd+1} with dilation $= d$ and the two roots mapped to adjacent hypercube nodes.

Basis For $k = 0$, the two-rooted hyperpyramid $\bar{P}(0, d)$, which consists entirely of the two root nodes, is mapped to adjacent nodes in hypercube Q_1:

$$f_0(0, \varepsilon) = 0 \text{ and } f_0(0', \varepsilon) = 1.$$

Induction Assume that there exists an embedding function f_n that satisfies the induction hypothesis. In order to embed a two-rooted hyperpyramid $\bar{P}(n + 1, d)$ into a hypercube $Q_{(n+1)d+1}$, we consider the hypercube $Q_{(n+1)d+1}$ to be composed of 2^d copies of hypercube Q_{nd+1}, labeled $0, 1, \cdots, 2^d - 1$. Apply f_n to the embedding of each two-rooted hyperpyramid $\bar{P}(n, d)$ into a hypercube Q_{nd+1}. We use a superscript to distinguish nodes of different two-rooted hyperpyramids $\bar{P}(n, d)$ mapped to distinct

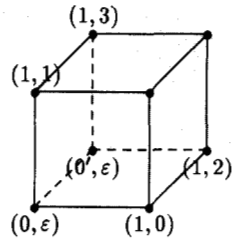

Figure 4

Two-rooted hyperpyramid $\tilde{P}(1, 2)$ embedded in hypercube Q_3, with dilation = 2.

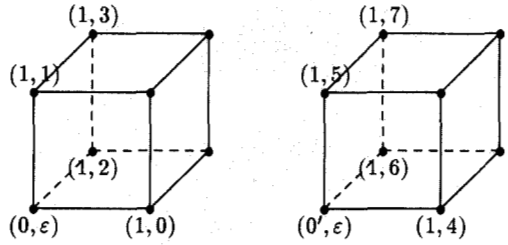

Figure 5

Two-rooted hyperpyramid $\tilde{P}(1, 3)$ embedded in hypercube Q_4, with dilation = 3.

hypercubes Q_{nd+1}. The following rules define the embedding function f_{n+1} in terms of f_n, for each hypercube, where $0 \le \ell < 2^d$ and j is a binary string of length $(i-1)d$.

$$f_{n+1}(0, \varepsilon) = f_n[(0, \varepsilon)^0],$$

$$f_{n+1}(0', \varepsilon) = f_n[(0, \varepsilon)^{2^{d-1}}],$$

$$f_{n+1}(1, \ell) = \begin{cases} f_n[(0', \varepsilon)^\ell] & \ell = 0 \text{ or } 2^{d-1}, \\ f_n[(0, \varepsilon)^\ell] & \text{otherwise,} \end{cases}$$

$$f_{n+1}(i, \ell | j) = f_n[(i-1, j)^\ell] \quad i > 1.$$

The first two equations define the two new roots. [The two roots can be chosen from the spare roots of any two adjacent hypercubes. We choose hypercubes 0 and 2^{d-1} thus; the two roots are mapped to hypercube addresses $(00\cdots0)$ and $(10\cdots0)$, respectively.] The third equation defines nodes at level 1. The last equation defines nodes at lower levels, where ℓ and j are binary strings of lengths d and id, respectively. **Figures 4** and **5** show the embeddings for $\tilde{P}(1, 2)$ and $\tilde{P}(1, 3)$, respectively.

Given any two binary strings x and y, let $\mathrm{HA}(x, y)$ be the Hamming distance between x and y, and let $W(x)$ denote the number of 1-bits (Hamming weight) of x; i.e., $W(x) = \mathrm{HA}(x, 0)$. For all $0 \le j < 2^d$ and $0 \le m < d$, we have the following properties:

1. $\mathrm{HA}[f_{n+1}(0, \varepsilon), f_{n+1}(1, j)] \le d$, demonstrated as follows:

$$\mathrm{HA}[f_{n+1}(0, \varepsilon), f_{n+1}(1, j)] = \begin{cases} \mathrm{HA}\{f_n[(0, \varepsilon)^0], f_n[(0, \varepsilon)^j]\} = W(j) < d & \text{if } j \ne 0 \text{ and } j \ne 2^{d-1}, \\ \mathrm{HA}\{f_n[(0, \varepsilon)^0], f_n[(0', \varepsilon)^0]\} = 1 & \text{if } j = 0, \\ \mathrm{HA}\{f_n[(0, \varepsilon)^0], f_n[(0', \varepsilon)^{2^{d-1}}]\} = 2 & \text{if } j = 2^{d-1}. \end{cases}$$

2. $\mathrm{HA}[f_{n+1}(0', \varepsilon), f_{n+1}(1, j)] \le d$: The proof follows the preceding proof.
3. $\mathrm{HA}[f_{n+1}(1, j), f_{n+1}(1, j^{\{m\}})] \le 2$: The distance is 1, except if $m \ne d-1$ and $j = 0$ or 2^{d-1}, for which the distance is 2.
4. $\mathrm{HA}[f_{n+1}(0, \varepsilon), f_{n+1}(0', \varepsilon)] = 1$.
5. The Hamming distance between corresponding nodes of adjacent hypercubes is 1.
6. The dilation of each edge in $\tilde{P}(n, d)$ is unchanged in the new embedding.

The induction hypothesis follows from these properties. ■

By substituting f_k recursively as defined by the induction rules, an explicit expression for f_k is obtained:

$$f_k(i, j) = \begin{cases} 0^{kd+1} & i = 0, \\ 10^{kd} & i = 0', \\ jx0^{(k-i)d} & 1 \le i \le k, \end{cases} \tag{1}$$

where $x = 1$, if $j_{d-2}\, j_{d-3} \cdots j_0 = 0$; and $x = 0$, otherwise. The expansion of the embedding function f_k is less than 2 (except for $k = 0$). **Figure 6** shows the hypercube addresses of the nodes of the hyperpyramid $\hat{P}(2, 2)$.

We now derive a lower bound on the dilation; however, we first need a proposition on the diameter of $\hat{P}(k, d)$.

Proposition 2 The diameter of $\hat{P}(k, d)$ for $d \ge 2$ is $2k$.

Proof Any two nodes x and y in $\hat{P}(k, d)$ are within a distance of $2k$, as one can define a path starting from x, traversing up to the root node and traversing down to y within $2k$ steps. Thus, the diameter is at most $2k$. We now show that the diameter is at least $2k$. Consider the two nodes $x = (k, 0^{kd})$ and $y = (k, 1^{kd})$, which are at the bottom level of $\hat{P}(k, d)$. Consider any path between x and y, and let h be the number of the highest level in $\hat{P}(k, d)$

that the path has touched. Clearly, the path must contain at least $2(k - h)$ edges in traversing up and down. Furthermore, there are hd hypercube dimensions that remain to be traversed, which requires at least hd edges. Thus the path has a length of at least $2(k - h) + hd = 2k + h(d - 2)$, which is minimized to $2k$ when $h = 0$ (recall that $d \geq 2$). ■

Proposition 3 A lower bound for the dilation of any embedding of a $\hat{P}(k, d)$ hyperpyramid into the smallest hypercube having enough nodes, Q_n, is $d/2$.

Proof From Proposition 2, the diameter of a hyperpyramid $\hat{P}(k, d)$ is $2k$. The smallest cube Q_n that is large enough to hold a hyperpyramid $\hat{P}(k, d)$ has $n = kd + 1$ dimensions. Since the hyperpyramid contains more than 2^{n-1} nodes, there exist two hyperpyramid nodes that are mapped to hypercube nodes at a distance of at least $n - 1$ in the hypercube Q_n. Consider any shortest path between these two hyperpyramid nodes. Let the length of the path be ℓ. Clearly, $\ell \leq 2k$. Edges on the path will be stretched in the embedding, so that all ℓ edges together are stretched into the path of length $\geq n - 1$ in the hypercube Q_n. Thus, at least one of these ℓ edges is stretched with dilation $\geq (n - 1)/\ell \geq (n - 1)/2k = d/2$. ■

Congestion

Here, we derive upper and lower bounds for the congestion. For the upper bound derivation, we need the two lemmas given next.

Lemma 1 There exists a spanning tree in Q_n such that each subtree of the root is of size at most $\lceil (2^n - 1)/n \rceil$.

Proof Such a spanning tree is constructed by modifying the spanning balanced n-tree in Q_n, denoted T, defined in [15, 16]. First, all cyclic nodes in T, which are all leaf nodes, are removed. The remaining tree, denoted T', has n subtrees isomorphic to one another. All the removed cyclic nodes are organized according to sets (*degenerated necklaces*) so that two nodes are in the same set if the address of one node can be derived by rotating the address of the other. Then, the cyclic nodes are added back to T', one set at a time in a round-robin manner, starting from subtree 0. It is easy to show that any degenerated set of k nodes can be added to any k consecutive subtrees (in a cyclic manner) in T' so that each added tree edge is also a hypercube edge. Thus, when all cyclic nodes are added back to T', each subtree has at most $\lceil (2^n - 1)/n \rceil$ nodes. ■

Lemma 2 A 2^n-node flat tree (i.e., a root with $2^n - 1$ children) can be embedded in a hypercube Q_n with congestion $\leq \lceil (2^n - 1)/n \rceil$.

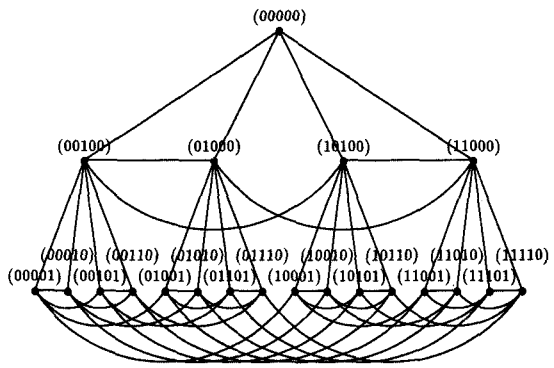

Figure 6

Hypercube addresses of the nodes of an embedded hyperpyramid $\hat{P}(2, 2)$.

Proof Denote the flat tree by T and the root of T by r. From Lemma 1, one can create a spanning tree, denoted T', in Q_n so that each subtree of the root has at most $\lceil (2^n - 1)/n \rceil$ nodes. Then, embed the flat tree T into the spanning tree T', which in turn is mapped to Q_n. Furthermore, stretch each edge (r, i) in T into a path corresponding to the path in traversing from node r toward node i in T'. Thus, the congestion of any edge in T is less than or equal to the maximum number of nodes in any subtree of the root in T', which is at most $\lceil (2^n - 1)/n \rceil$. ■

We are now ready to give an upper bound on the congestion.

Theorem 2 An upper bound of the congestion for an embedding of $\hat{P}(k, d)$ with dilation $= d$ into Q_{kd+1}, for $d \geq 2$, is $\lceil (2^d - 1)/d \rceil$.

Outline of proof The proof can be performed by induction based on the following arguments. The maximum edge congestion is caused by the hyperpyramid edges between the root node and its 2^d children. (Note that in considering the congestion, we need not consider the spare root.) Among the 2^d children, $2^d - 2$ are in a hypercube Q_d. The other two children are neighbors of the two roots but are not contained in the hypercube Q_d. The two roots are in the same hypercube Q_d as the $2^d - 2$ children. By Lemma 2, the congestion caused by the edges between the real root and its children in the hypercube Q_d is bounded from above by $\lceil (2^d - 1)/d \rceil$. We route the

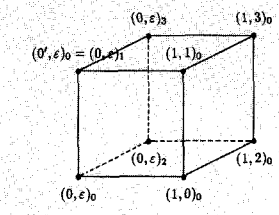

Figure 7

Basis for induction in Theorem 3: A two-rooted hyperpyramid $\tilde{P}_0(1, 2)$, a hyperpyamid $\hat{P}_2(0, 2)$, and hyperpyramid $\hat{P}_3(0, 2)$ embedded in hypercube Q_3, with dilation = 3.

$d - 1$ paths with length = 2 from node $(1, 0)$ or $(1, 2^{d-1})$ to its $d - 1$ neighbors through an unused hypercube node. The path between nodes $(0, \varepsilon)$ and $(1, 2^{d-1})$ is routed through node $(1, 0)$. Note that the congestion of the edges in the hypercube Q_d does not increase during the next induction step. ■

A simple lower bound can be derived as follows.

Proposition 4 A lower bound of the congestion for any embedding of a hyperpyramid $\hat{P}(k, d)$ into a hypercube Q_{kd+1} is $1 + \lceil (2^d - d)/(kd + 1) \rceil$.

Proof The nodes at level $k - 1$ of a hyperpyramid $\hat{P}(k, d)$ have degree $1 + (k - 1)d + 2^d$. The degree of a hypercube Q_{kd+1} is $kd + 1$. Thus, a lower bound of the congestion is

$$\left\lceil \frac{1 + (k-1)d + 2^d}{kd + 1} \right\rceil = 1 + \left\lceil \frac{2^d - d}{kd + 1} \right\rceil. \quad \blacksquare$$

As a corollary of Theorem 2, we have an embedding of $\hat{P}(k, 2)$ into Q_{2k+1} with dilation = 2 and congestion = 2. As a comparison, the embeddings of pyramid $P(k, 2, 2)$, which is a subgraph of hyperpyramid $\hat{P}(k, 2)$, into Q_{2k+1} given in [26] have dilation = 2 and congestion = 3, or alternatively, dilation = 3 and congestion = 2.

• *Embedding multiple hyperpyramids into a hypercube*

Even though minimal expansion (i.e., expansion <2) is achieved in the embedding described in the preceding subsection, $2^d - 2$ hypercube nodes are not used in each induction step. It is possible, however, to embed a $\hat{P}(k, d)$ hyperpyramid and $2^d - 2$ smaller hyperpyramids $\hat{P}(k - 1, d)$ into a hypercube Q_{kd+1} at the same time, so that only one hypercube node is not used. In this subsection, we present such an embedding with dilation = $d + 1$ and congestion = $2^d + \lceil (2^d - 1)/d \rceil + 1$. As before, the inductive construction given in the next theorem is mainly required for bounding the congestion later.

Dilation

Theorem 3 A hyperpyramid $\hat{P}(k, d)$ together with $2^d - 2$ hyperpyramids $\hat{P}(k - 1, d)$, $k \geq 1$ and $d \geq 2$, can be embedded in Q_{kd+1} with expansion ≈1 (only one hypercube node is not used) and dilation = $d + 1$.

Proof In the following, the subscripts on $\tilde{P}$, $\hat{P}$, and the node identifiers are used to identify different hyperpyramids and nodes therein. For notational convenience, we let $(0, \varepsilon)_1$ denote $(0', \varepsilon)_0$. For the proof, we consider a two-rooted hyperpyramid $\tilde{P}(k, d)$ and $2^d - 2$ hyperpyramids $\hat{P}(k - 1, d)$ (with single roots). Let the embedding function be f_k. The proof is by induction, and the hypothesis is that the following two conditions hold for $1 \leq k \leq n$:

1. A two-rooted hyperpyramid $\tilde{P}_0(k, d)$ and $2^d - 2$ hyperpyramids $\hat{P}_j(k - 1, d)$, $2 \leq j < 2^d$, $k \geq 1$ and $d \geq 2$, can be embedded in a Q_{kd+1} hypercube, with dilation = $d + 1$.
2. $\mathrm{HA}\{f_k[(0, \varepsilon)_x], f_k[(0, \varepsilon)_{x\{m\}}]\} = 1$ for all $0 \leq x < 2^d$, $0 \leq m < d$; i.e., all of the 2^d roots are mapped to a subcube Q_d in the hypercube Q_{kd+1}, and the two roots of $\tilde{P}_0$ are mapped to adjacent hypercube nodes.

Basis For $k = 1$, the two-rooted hyperpyramid $\tilde{P}_0(1, d)$ contains the roots $(0, \varepsilon)_0$ and $(0, \varepsilon)_1$, and the base Q_d $[(1, j)_0, 0 \leq j < 2^d]$. For each of the $2^d - 2$ hyperpyramids $\hat{P}_x(0, d)$, $x \in \{2, 3, \cdots, 2^d - 1\}$, $\hat{P}_x(0, d)$ is the root node. Define f_1 as

$$f_1[(0, \varepsilon)_j] = j|0 \qquad 0 \leq j < 2^d,$$

$$f_1[(1, j)_0] = j|1 \qquad 0 \leq j < 2^d.$$

It is easily seen that f_1 satisfies the two conditions of the hypothesis. **Figures 7** and **8** show the embedding for $k = 1$, with $d = 2$ and $d = 3$, respectively.

Induction Assume that the mapping f_n satisfies the above two conditions. Consider a hypercube $Q_{(n+1)d+1}$ with hyperpyramids embedded by the function f_n in each of the 2^d copies of hypercube Q_{nd+1}. We define f_{n+1} in terms of f_n by the following rules, where $0 \leq \ell < 2^d$ and j is a binary string of length $(i - 1)d$:

R1: $f_{n+1}[(0, \varepsilon)_\ell] = f_n[(0, \varepsilon)_0^\ell]$,

R2: $f_{n+1}[(1, \ell)_0] = f_n[(0, \varepsilon)_1^\ell]$,

R3: $f_{n+1}[(i, \ell|j)_0] = f_n[(i-1, j)_0^{\ell}] \qquad 2 \le i \le n+1,$

R4: $f_{n+1}[(i, \ell|j)_x] = f_n[(i-1, j)^{\ell}_{x \oplus \gamma(\ell, x)}] \qquad 1 \le i \le n+1, \quad 2 \le x < 2^d.$

The symbol $\oplus$ represents the bitwise exclusive or. The d-bit string $\gamma(\ell, x) = (\gamma_{d-1}\ \gamma_{d-2} \cdots \gamma_0)$ is determined from ℓ and x as follows: for $0 \le m < d-1$, if $\ell_m = 0$ and $x_{d-1}\ x_{d-2} \cdots x_{m+1} \neq 0$, then $\gamma_m = 1$; otherwise $\gamma_m = 0$. The superscript ℓ identifies nodes of different hyperpyramids mapped to distinct hypercubes Q_{nd+1}, as before. Thus, $f_n[(i, j)_x^{\ell}] = \ell | f_n[(i, j)_x]$. By R1 of the recursive definition above, we select the root $(0, \varepsilon)_0$ of $\bar{P}_0(n, d)$ to be the spare root. In the induction, 2^d hypercubes with embedded hyperpyramids are used to form a new embedding. The number of spare roots in the 2^d hypercubes is 2^d. Two of them serve as the two new roots of $\bar{P}_0(n+1, d)$, and the remaining $2^d - 2$ spare roots serve as the new roots of the $2^d - 2$ hyperpyramids $\hat{P}(n, d)$, one for each. [For notational convenience, we choose the two new roots of $\bar{P}(n+1, d)$ from hypercubes 0 and 1, instead of choosing from hypercubes 0 and 2^{d-1} as in the subsection on embedding a hyperpyramid into a hypercube.] By R2, we select $(0, \varepsilon)_1$ as the real root of the two-rooted hyperpyramid $\bar{P}_0(n, d)$ in each hypercube Q_{nd+1}; i.e., it becomes a node at level 1 of the hyperpyramid $\bar{P}_0(n+1, d)$. R3 moves nodes of $\bar{P}_0(n, d)$ at level $i-1$ to nodes of $\bar{P}_0(n+1, d)$ at levels $i \ge 2$. R4 moves nodes of the hyperpyramids $\hat{P}_x(n-1, d)$, $2 \le x < 2^d$, at level $i-1$ to nodes of the hyperpyramids $\hat{P}_x(n, d)$ at level i. Note that R4 is complicated by the exchange between adjacent hyperpyramids as defined by γ. For example, for $d = 3$ and $\ell = 0$, $\gamma = 001, 001, 011, 011, 011, 011$ for $x = 2, 3, 4, 5, 6, 7$, respectively. The naive embedding without exchange, i.e., $\gamma = 0$, would have dilation $= 2d$ for some hyperpyramid.

Figure 9 shows the induction step of the naive embedding ($\gamma = 0$), for $d = 3$. In general, the dilation ranges from $d+1$ to $2d$, depending on the hyperpyramid. With the exchanges defined by γ, the embedding is shown in **Figure 10**. The exchange is indicated by two-way arrows.

We now prove that the recursive definition is "well-defined," by which we mean that if $f_{n+1}[(i, \ell|j)_x] = f_{n+1}[(i', \ell'|j')_{x'}]$, then $x = x'$, $i = i'$, $\ell = \ell'$, and $j = j'$. This is obvious if $\gamma(\ell, x) \equiv 0$. With γ a nonzero function of ℓ and x, it suffices to prove that $f_{n+1}[(i, \ell|j)_{x \oplus \gamma(\ell, x)}] = f_n[(i-1, j)_x^{\ell}]$. From R4, we have $f_{n+1}[(i, \ell|j)_{x \oplus \gamma(\ell, x)}] = f_n\{(i-1, j)^{\ell}_{x \oplus \gamma(l, x) \oplus \gamma[\ell, x \oplus \gamma(\ell, x)]}\}$. Thus, we simply prove that $\gamma(\ell, x) = \gamma[\ell, x \oplus \gamma(\ell, x)]$. This is true by Lemma 3, shown later.

We now prove that the recursive definition satisfies the induction hypotheses. Condition 2 of the induction hypotheses is preserved because of R1 in the definition of

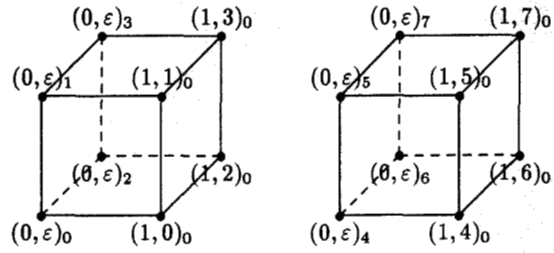

Figure 8

Basis for induction in Theorem 3: A two-rooted hyperpyramid $\bar{P}_0(1, 3)$ and six hyperpyramids $\hat{P}_j(0, 3)$, $j \in \{2, 3, \cdots, 7\}$, embedded in hypercube Q_4, with dilation $= 4$. [Note that node $(0, \varepsilon)_1$ is identical to node $(0', \varepsilon)_0$.]

f_{n+1}. In order to prove that condition 1 holds for $k = n+1$, we partition the newly formed hyperpyramid edges into three disjoint sets, S_1, S_2, and S_3, by a definition similar to the one used in the proof of Theorem 3. First, the dilation of edges in S_3 is preserved. We prove that the dilation of edges in S_2 is either 1 or 2 by considering the Hamming distance between $f_{n+1}[(i, \ell|j)_x]$ and $f_{n+1}[(i, \ell^{\{m\}}|j)_x]$:

$$f_{n+1}[(i, \ell|j)_x] = f_n[(i-1, j)^{\ell}_{x \oplus \gamma(\ell, x)}]$$

$$= \ell | f_n[(i-1, j)_{x \oplus \gamma(\ell, x)}].$$

$$f_{n+1}[(i, \ell^{\{m\}}|j)_x] = f_n[(i-1, j)^{\ell^{\{m\}}}_{x \oplus \gamma(\ell^{\{m\}}, x)}]$$

$$= \ell^{\{m\}} | f_n[(i-1, j)_{x \oplus \gamma(\ell^{\{m\}}, x)}].$$

Let $\gamma(\ell, x) = y$. Then, from the definition of $\gamma(\ell, x)$, one can derive $\gamma(\ell^{\{m\}}, x) = y$ or $y^{\{m\}}$. Thus, the dilation in S_2 is either 1 or 2. A dilation of 2 occurs when there is an exchange operation involved in one side of the hypercubes. To determine the edge dilation in S_1, we consider subsets S_{11}, the edges between nodes at level 1, and S_{12}, the edges between nodes at level 1 and the roots. The edge dilation in S_{11} is either 1 or 2, for the same reasons the dilation of edges in the set S_2 is at most 2. For the edge dilation in S_{12}, consider $\mathrm{HA}\{f_{n+1}[(0, \varepsilon)_x], f_{n+1}[(1, \ell)_x]\}$, which is $W(\ell) + 1 \le d+1$, if $x = 0$ or 1. For $x \neq 0$ and $x \neq 1$, $f_{n+1}[(0, \varepsilon)_x] = f_n[(0, \varepsilon)_0^x]$, and $f_{n+1}[(1, \ell)_x] = f_n[(0, \varepsilon)^{\ell}_{x \oplus \gamma(\ell, x)}]$. Thus, the Hamming distance is $W[x \oplus \gamma(\ell, x)] + W(x \oplus \ell)$, which is at most $d+1$ by Lemma 4, shown later. ■

To complete the proof of the above theorem, we prove the next two lemmas, which were used in the theorem.

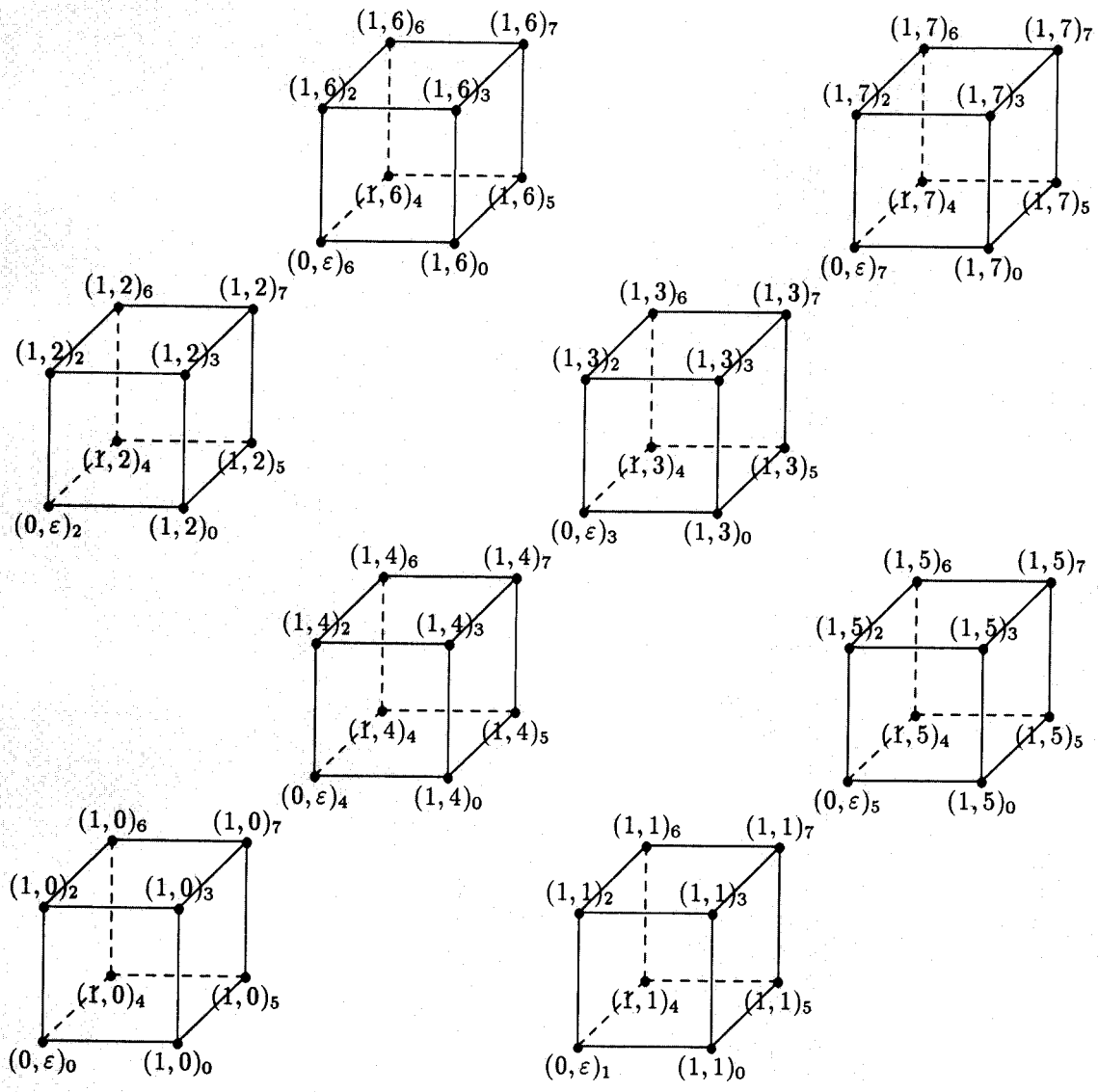

Figure 9

Induction step of the naive embedding ($\gamma = 0$), for $d = 3$. The dilation is 6.

Lemma 3 $\gamma(\ell, x) = \gamma[\ell, x \oplus \gamma(\ell, x)]$.

Proof Let $y = \gamma(\ell, x)$ and $\gamma' = \gamma(\ell, x \oplus y)$. Then, we must prove that $\gamma'_m = y_m$ for all $0 \le m \le d - 1$. Let $x' = x \oplus y$. From the definition of γ in the proof of Theorem 3, we have the following:

- If $\ell_m = 1$, then $y_m = \gamma'_m = 0$.
- If $\ell_m = 0$:
  - If $(x_{d-1}x_{d-2} \cdots x_{m+1}) = 0$, then $y_m = 0$ and $(y_{d-1}\, y_{d-2} \cdots y_{m+1}) = 0$. Thus, $(x'_{d-1}\, x'_{d-2} \cdots x'_{m+1}) = 0$; i.e., $\gamma'_m = 0$. Therefore, $y_m = \gamma'_m$.
  - If $(x_{d-1}x_{d-2} \cdots x_{m+1}) \neq 0$, then $y_m = 1$. Let x_r be the leading nonzero bit of x, where $m + 1 \le r \le d - 1$. Then, $y_r = 0$, and $x'_r = x_r \oplus y_r = 1$; i.e., $(x'_{d-1}x'_{d-2} \cdots x'_{m+1}) \neq 0$. Thus, $\gamma'_m = 1$. Therefore, $y_m = \gamma'_m$. ■

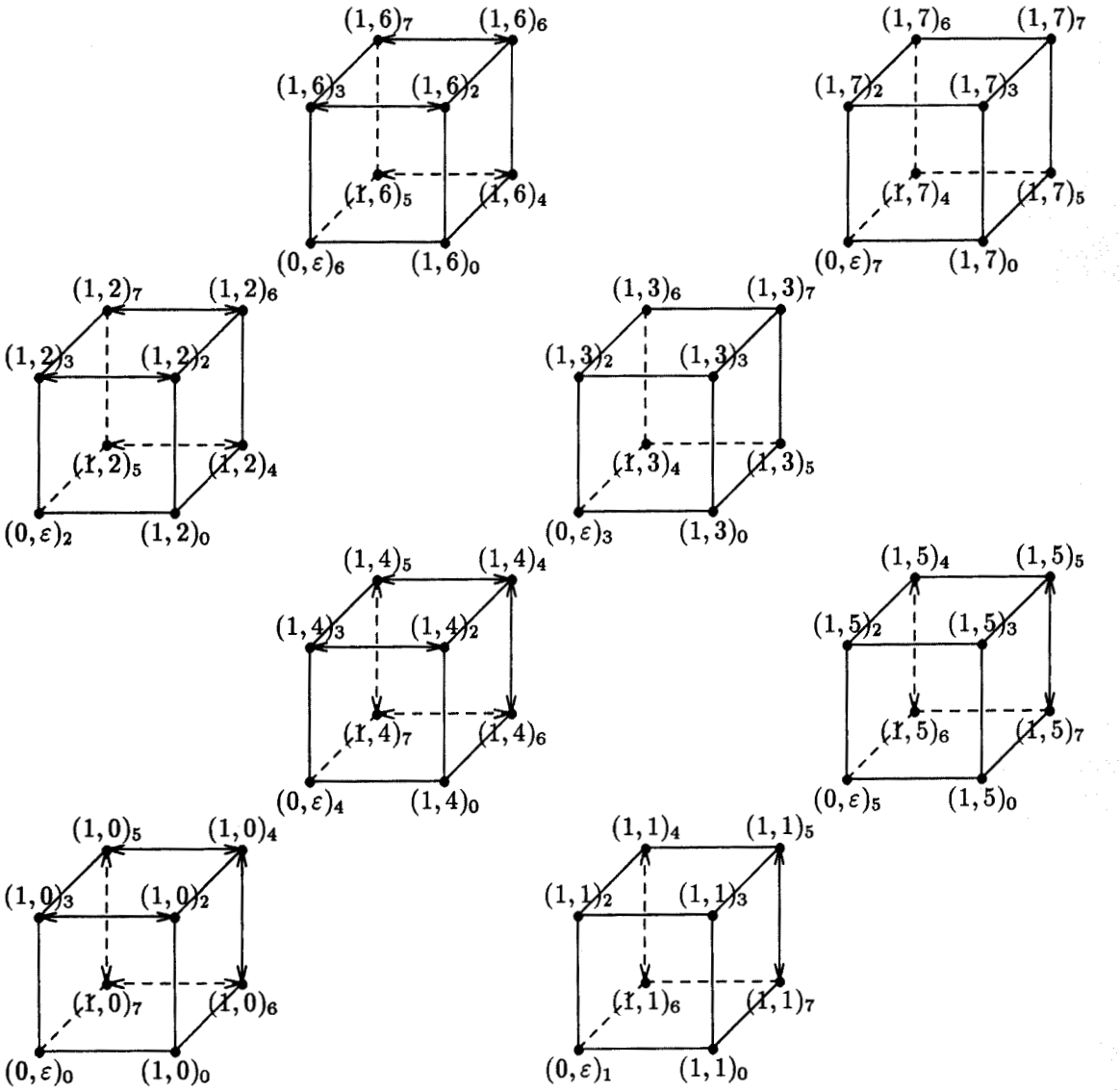

Figure 10

"Improved" embedding by performing an exchange described by γ of the induction step, for $d = 3$, with dilation 4.

Lemma 4 $W[x \oplus \gamma(\ell, x)] + W(x \oplus \ell) \leq d + 1$, where $2 \leq x < 2^d$, $0 \leq \ell < 2^d$.

Proof We prove this lemma by showing that

$$\sum_{m=0}^{d-1} [(x_m \oplus \gamma_m) + (x_m \oplus \ell_m)] \leq d + 1.$$

Let x_r be the leading nonzero bit of x ($r = -1$, if $x = 0$). Consider any m such that $x_m \oplus \ell_m = 1$. There are three cases:

- $m < r$: If $x_m = 0$, then $\ell_m = 1$ and $\gamma_m = 0$. If $x_m = 1$, then $\ell_m = 0$ and $\gamma_m = 1$. For both cases, $(x_m \oplus \gamma_m) + (x_m \oplus \ell_m) = 1$.

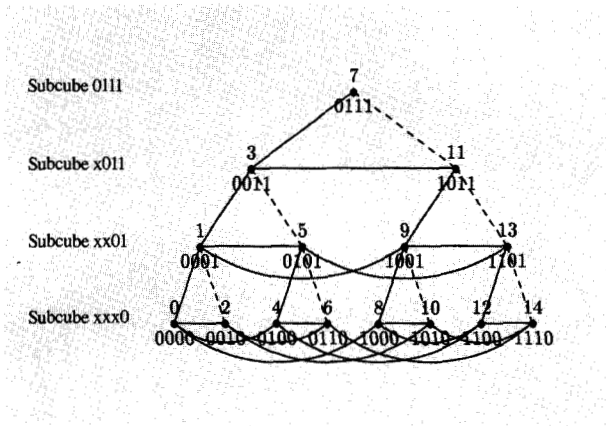

Figure 11

Topology of hyperpyramid $\hat{P}(3, 1)$. The dashed lines represent edges of dilation 2.

- $m = r$: $x_m = x_r = 1$. Thus, $\ell_m = 0$ and $\gamma_m = 0$ (since $x_{d-1}x_{d-2} \cdots x_{m+1} = 0$). We have $(x_m \oplus \gamma_m) + (x_m \oplus \ell_m) = 2$.
- $m > r$: $x_m = 0$, since x_r is the leading nonzero bit. Then, $\ell_m = 1$, and $\gamma_m = 0$. We have $(x_m \oplus \gamma_m) + (x_m \oplus \ell_m) = 1$.

In summary, for any m such that $x_m \oplus \ell_m = 1$, we have $(x_m \oplus \gamma_m) + (x_m \oplus \ell_m) = 1$, except for $m = r$, for which $(x_m \oplus \gamma_m) + (x_m \oplus \ell_m) = 2$. For any m such that $x_m \oplus \ell_m = 0$, we have $(x_m \oplus \gamma_m) + (x_m \oplus \ell_m) \leq 1$. Therefore,

$$\sum_{m=0}^{d-1} [(x_m \oplus \gamma_m) + (x_m \oplus \ell_m)] \leq d + 1. \quad \blacksquare$$

Congestion

We now show that the congestion for the embedding defined in Theorem 3 is at most $2^d + \lceil (2^d - 1)/d \rceil + 1$. First, we need the following lemma.

Lemma 5 A 2^n-node complete graph with all edges duplicated can be embedded into a hypercube Q_n, with congestion equal to 2^n.

Proof[1] Since all edges in the complete graph are duplicated, one can decompose the edges of the complete graph into sets E_i, for $0 \leq i < 2^n$, so that the graph $G_i = (V, E_i)$ forms a 2^n-node flat tree rooted at node i. (V is the node set of the complete graph.) One can embed the flat tree G_i into the spanning binomial tree (SBT) [15] rooted at node i in Q_n. (A spanning binomial tree rooted at node i in Q_n is constructed by running the well-known recursive doubling algorithm spanning hypercube dimensions in the order $0, 1, \cdots, n - 1$.) It is easy to show that any edge in dimension j has 2^{j+1} spanning trees passing through it, and the subtree connected through the edge in each spanning tree is of size 2^{n-1-j}. Thus, the congestion of any edge in dimension j is $2^{j+1} \times 2^{n-1-j} = 2^n$. $\blacksquare$

Theorem 4 The congestion for the embedding in Theorem 3 with dilation $= d + 1$ is $2^d + \lceil (2^d - 1)/d \rceil + 1$.

Outline of proof For the sake of brevity, we give an outline of an inductive proof. We let the path from $(0, \varepsilon)_i$ to $(1, j)_i$ pass through an intermediate node $(0, \varepsilon)_j$. For example, in Figure 10, the path from $(0, \varepsilon)_3$ to $(1, 4)_3$ is defined to go through the intermediate node $(0, \varepsilon)_4$. For convenience, define S_1 to be the set of paths from each node $(0, \varepsilon)_i$ to each of its 2^d intermediate nodes $(0, \varepsilon)_j$. Also define S_2 to be the set of paths from each intermediate node $(0, \varepsilon)_j$ to each node $(1, j)_i$. Note that there are 2^d roots, which have the form $(0, \varepsilon)_i$. The 2^d roots are the same set of nodes as the 2^d intermediate nodes. The congestion of edges in S_1 is the same as that encountered when embedding 2^d flat trees, each of size 2^d and rooted at a different node, in a hypercube Q_d. By Lemma 5, this congestion is bounded from above by 2^d. For all edges in S_2, the congestion is the same as that encountered when embedding a single flat tree in a hypercube Q_d, which is bounded from above by $\lceil (2^d - 1)/d \rceil$ according to Lemma 2. The inductive hypothesis is that the congestion of the edges in the new d hypercube dimensions (i.e., the paths in S_1) is at most 2^d. During an induction step, congestions of the edges of the d hypercube dimensions under consideration will increase by at most $\lceil (2^d - 1)/d \rceil + 1$ (which accounts for the paths in S_2 plus the edge introduced by the exchange operation γ). Note that the path assignment for the basis (see for example Figure 8) can be done such that the edge congestion is 1 for edges in dimension 0, and is at most $\lceil (2^d - 1)/d \rceil$ for edges in dimensions 1 to d. $\blacksquare$

Note that it is possible to have an embedding of dilation = 2d with congestion $= O(2^d/d)$ [31]. Also, for $d = 2$, it is possible to achieve an embedding of one hyperpyramid $\hat{P}(k, 2)$ and two smaller hyperpyramids $\hat{P}(k - 1, 2)$ with dilation = 3 and congestion $\doteq$ 3 [31], by fine-tuning the path assignments in the induction step. This improves the result in [27], which has dilation = 3 and congestion = 6.

- *Remarks*

When we include the case $d = 1$ in Theorem 1, the theorem becomes the following: A hyperpyramid $\hat{P}(k, d)$ can be embedded in a hypercube Q_{kd+1} with dilation =

[1] Peter F. Corbett, personal communication, IBM Research Division, October 7, 1991.

$\max(d, 2)$. As a corollary of this, a hyperpyramid $\hat{P}(k, 1)$ can be embedded in a hypercube Q_{k+1} with dilation = 2. **Figure 11** shows a hyperpyramid $\hat{P}(3, 1)$. Note that an X-tree [32] is isomorphic to a pyramid $P(k, 2, 1)$, which in turn is a subgraph of a hyperpyramid $\hat{P}(k, 1)$, by Proposition 1. (**Figure 12** shows an example of a three-level X-tree.) Thus, an X-tree can be embedded in a hypercube with expansion < 2 and dilation = 2.

Since a hyperpyramid $\hat{P}(k, 1)$ contains a complete binary tree as a subgraph, our result degenerates to the following: A complete binary tree can be embedded in a hypercube with expansion ≈ 1 and dilation = 2. This result was first discovered by Nebeský [9] and rediscovered independently in [11], [12], and [14]. All embeddings except the one in [11] also guarantee that only one of the tree edges is of dilation = 2. Our method is the same as that of [11], in which the edge to the left child of every nonleaf node is of dilation = 1 and the edge to the right child is of dilation = 2. However, in our embedding and the embedding in [11], all nodes at the same level form a subcube and therefore have additional adjacencies (e.g., Figure 11). Our embedding and the embedding in [11] are equivalent to labeling a complete binary tree according to an "inorder" traversal [33] with a starting index of 0 or 1. Such an embedding was also used in [3, 10].

Notice that an embedding of an X-tree with dilation = 2 can also be obtained by an inorder traversal, by interpreting the label as a binary-reflected Gray code [34], as observed by Bhatt[2], e.g., Figure 12. (This is because two binary-reflected Gray codes with a power of 2 difference in their addresses are at most Hamming distance 2 apart [34].) However, the number of edges with dilation = 2 is higher for such an embedding than for our embedding.

When the hypercube connections at level i are ignored for $0 \le i \le k$, the hyperpyramid $\hat{P}(k, d)$ becomes a k-level complete (2^d)-ary tree. A corollary of Theorem 1 is that a k-level complete n-ary tree can be embedded in a hypercube with dilation = $\max(2, \lceil \log_2 n \rceil)$ and expansion = $(2^{k\lceil \log_2 n \rceil + 1})(n - 1)/(n^{k+1} - 1)$. The expansion is less than 2 when n is a power of 2. The previous result by Wu [11] has dilation = $2\lceil \log_2 n \rceil$. Similarly, a corollary of Theorem 3 is that a k-level complete n-ary tree together with $2^{k\lceil \log_2 n \rceil} - 2$ complete n-ary trees of level $k - 1$ can be embedded in a hypercube of dimension $k\lceil \log_2 n \rceil + 1$ with dilation = $\lceil \log_2 n \rceil + 1$. The expansion is approximately 1 when n is a power of 2.

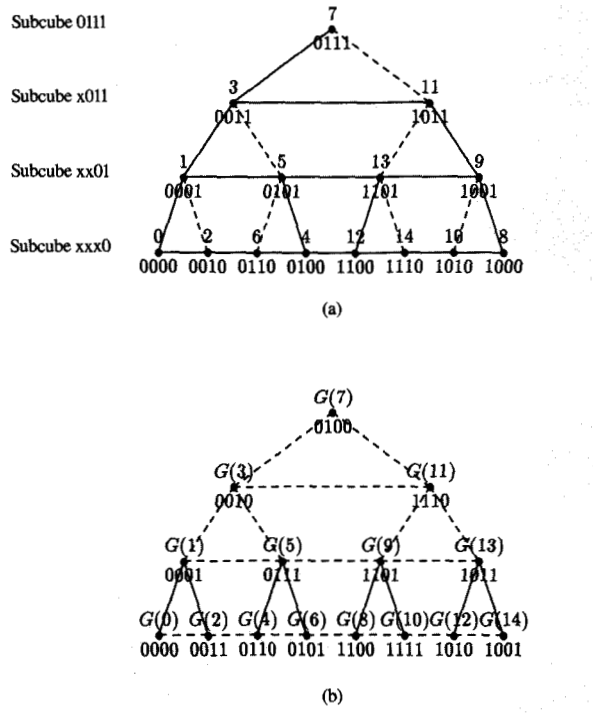

Figure 12

Topology of a three-level X-tree. The dashed lines represent edges of dilation 2. (a) The embedding is derived by an "inorder" traversal. (b) The embedding is derived by interpreting the label in part (a) as a binary-reflected Gray code.

Summary

We have presented embeddings from pyramids (the guest graph) into hypercubes (the host graph) with minimal expansion, dilation = 2, and congestion = 2. We have also described embeddings from hyperpyramids (the guest graph), i.e., pyramids in which each nonleaf node has 2^d children interconnected as hypercubes, into hypercubes (the host graph) with minimal expansion and dilation = d. The congestion is bounded from below by $1 + \lceil (2^d - d)/(kd + 1) \rceil$ and from above by $\lceil (2^d - 1)/d \rceil$.

The expansion is asymptotically 1.5 for the embedding of pyramid $P(k, 2, 2)$, and 2 for the embedding of hyperpyramid $\hat{P}(k, d)$. In the first case, about a third of the hypercube nodes are unused, and in the second, about half of them are unused. When two pyramids of height $k - 1$ are embedded together with a pyramid of height k, the expansion becomes approximately 1. Lai and White [27] described such an embedding with dilation = 3 and congestion = 6. We improved it to dilation = 3 and congestion = 3. We then generalized it to an embedding of $2^d - 2$ hyperpyramids of height $k - 1$ together with a hyperpyramid of height k into a $(kd + 1)$-dimensional hypercube. Only one hypercube node is not used in such an embedding, the dilation is $d + 1$, and the congestion is $O(2^d)$.

[2] Sandeep N. Bhatt, personal communication, Dept. of Computer Science, Yale University, New Haven, CT, 1987.

It follows from the hyperpyramid embeddings that a pyramid $P(k, 2^i, 2^{d-i})$ can be embedded in a hypercube with minimal expansion, dilation = d, and a congestion of at most $\lceil (2^d - 1)/d \rceil$. A pyramid and $2^d - 2$ smaller pyramids $P(k, 2^i, 2^{d-i})$ (possibly different i's for different pyramids) can be embedded in a hypercube with minimal expansion, dilation = $d + 1$, and congestion of at most $O(2^d)$. The congestion can be reduced by a factor of d if the dilation is increased to $2d$. A complete n-ary tree can be embedded in a hypercube with minimal expansion and dilation = $\max(2, \lceil \log_2 n \rceil)$ when n is a power of 2. The best embedding known previously has dilation = $2\lceil \log_2 n \rceil$ [11]. Our results also provide embeddings of degenerate hyperpyramids, such as complete binary trees and X-trees, with minimal expansion and dilation = 2.

Acknowledgment

We thank David Greenberg for his comments on an early draft of this paper, and Peter Corbett, who provided the proof of Lemma 5 and improved, by a constant factor, the upper bound of the congestion from our original congestion. The generous support by the Office of Naval Research under Contract No. N00014-86-K-0310 is gratefully acknowledged.

References

1. Arnold L. Rosenberg and Larry Snyder, "Bounds on the Costs of Data Encodings," *Math. Syst. Theory* **12,** 9–39 (1978).
2. S. Lennart Johnsson and Peggy Li, "Solution Set for AMA/CS 146," *Technical Report 5085:DF:83*, California Institute of Technology, Pasadena, May 1983.
3. S. Lennart Johnsson, "Communication Efficient Basic Linear Algebra Computations on Hypercube Architectures," *J. Parallel Distr. Computing* **4,** 133–172 (April 1987).
4. M. Y. Chan and F. Y. L. Chin, "On Embedding Rectangular Grids in Hypercubes," *IEEE Trans. Computers* **C-37,** 1285–1288 (1988).
5. Ching-Tien Ho and S. Lennart Johnsson, "On the Embedding of Arbitrary Meshes in Boolean Cubes with Expansion Two Dilation Two," *Proceedings of the 1987 International Conference on Parallel Processing*, IEEE Computer Society, 1987, pp. 188–191.
6. David S. Greenberg, "Minimum Expansion Embeddings of Meshes in Hypercubes," *Technical Report YALEU/DCS/RR-535*, Dept. of Computer Science, Yale University, New Haven, CT, August 1987.
7. M. Y. Chan, "Dilation-2 Embeddings of Grids into Hypercubes," *Technical Report UTDCS 1-88*, Computer Science Dept., University of Texas at Dallas, 1988.
8. Ivan Havel and Petr Liebl, "Embedding the Polytomic Tree into the n-Cube," *Časopis pro Pěstování matematiky* (in Czechoslovakian) **98,** 307–314 (1973).
9. Ladislav Nebeský, "On Cubes and Dichotomic Trees," Časopis pro Pěstování matematiky (in Czechoslovakian) **99,** 164–167 (1974).
10. S. Lennart Johnsson, "Odd–Even Cyclic Reduction on Ensemble Architectures and the Solution of Tridiagonal Systems of Equations," *Technical Report YALE/DCS/RR-339*, Dept. of Computer Science, Yale University, New Haven, CT, October 1984.
11. Angela Y. Wu, "Embedding of Tree Networks in Hypercubes," *J. Parallel & Distr. Computing* **2,** 238–249 (1985).
12. Sandeep N. Bhatt and Ilse I. F. Ipsen, "How to Embed Trees in Hypercubes," *Technical Report YALEU/CSD/RR-443*, Dept. of Computer Science, Yale University, New Haven, CT, December 1985.
13. Sandeep N. Bhatt, Fan R. K. Chung, F. Tom Leighton, and Arnold L. Rosenberg, "Optimal Simulations of Tree Machines," *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science*, IEEE Computer Society, 1986, pp. 274–282.
14. Sanjay R. Deshpande and Roy M. Jenevin, "Scalability of a Binary Tree on a Hypercube," *Proceedings of the 1986 International Conference on Parallel Processing*, IEEE Computer Society, 1986, p. 661–668.
15. S. Lennart Johnsson and Ching-Tien Ho, "Spanning Graphs for Optimum Broadcasting and Personalized Communication in Hypercubes," *IEEE Trans. Computers* **38,** 1249–1268 (September 1989).
16. Ching-Tien Ho and S. Lennart Johnsson, "Spanning Balanced Trees in Boolean Cubes," *SIAM J. Sci. Stat. Comp.* **10,** 607–630 (July 1989).
17. Alan S. Wagner,"Embedding Trees in the Hypercube," Ph.D. thesis, University of Toronto, Ontario, 1987.
18. Marilynn Livingston and Quentin F. Stout, "Embeddings in Hypercubes," *Proceedings of the Sixth International Conference on Mathematical Modelling*, vol. 11, pp. 222–227, 1988.
19. Quentin F. Stout, "Sorting, Merging, Selecting, and Filtering on Tree and Pyramid Machines," *Proceedings of the 1983 International Conference on Parallel Processing*, IEEE Computer Society, 1983, pp. 214–221.
20. Russ Miller and Quentin F. Stout, "Data Movement Techniques for the Pyramid Computer," *SIAM J. Comput.* **16,** 38–60 (February 1987).
21. J. H. Chang, O. H. Ibarra, T. C. Pong, and S. M. Sohn, "Two-Dimensional Convolution on a Pyramid Computer," *Proceedings of the 1987 International Conference on Parallel Processing,* IEEE Computer Society, 1987, pp. 780–782.
22. *Pyramidal Systems for Computer Vision*, V. Cantoni and S. Levialdi, Eds., Springer-Verlag, Berlin, 1986.
23. Leonard Uhr, "Parallel, Hierarchical Software/Hardware Pyramid Architectures," *Pyramidal Systems for Computer Vision*, V. Cantoni and S. Levialdi, Eds., Springer-Verlag, Berlin, 1986, pp. 1–20.
24. Tony F. Chan and Yousef Saad, "Multigrid Algorithms on the Hypercube Multiprocessor," *IEEE Trans. Computers* **35,** 969–977 (November 1986).
25. Quentin F. Stout, "Hypercubes and Pyramids," *Pyramidal Systems for Computer Vision*, V. Cantoni and S. Levialdi, Eds., Springer-Verlag, Berlin, 1986, pp. 75–90.
26. Ten-Hwang Lai and William White, "Embedding Pyramids in Hypercubes," technical report, Dept. of Computer and Information Science, Ohio State University, Columbus, November 1987.
27. Ten-Hwang Lai and William White, "Mapping Multiple Pyramids into Hypercubes Using Unit Expansion," technical report, Dept. of Computer and Information Science, Ohio State University, Columbus, January 1988.
28. Sotirios G. Ziavras, "Efficient Mapping Algorithms for a Class of Hierarchical Systems," *IEEE Trans. Parallel & Dist. Syst.* **4,** No. 11, 1230–1245 (November 1993).
29. Sotirios G. Ziavras and Muhammad A. Siddiqui, "Pyramid Mappings onto Hypercubes for Computer Vision: Connection Machine Comparative Study," *J. Concurrency: Pract. & Exper.* **5,** 471–489 (September 1993).

30. Sandeep N. Bhatt and Charles E. Leiserson, *How to Assemble Tree Machines*, vol. 2, JAI Press Inc., 1984, pp. 95–114.
31. Ching-Tien Ho and S. Lennart Johnsson, "Embedding Hyper-pyramids into Hypercubes," *Technical Report YALEU/DCS/RR-667*, Department of Computer Science, Yale University, New Haven, CT, December 1988.
32. A. M. Despain and D. A. Patterson, "X-Tree—A Tree Structured Multiprocessor Architecture," *Proceedings of the 5th Symposium on Computer Architecture*, 1978, pp. 144–151.
33. Donald E. Knuth, *The Art of Computer Programming, Vol. 1: Fundamental Algorithms*, Addison-Wesley Publishing Co., Reading MA, 1968, pp. 315–317.
34. E. M. Reingold, J. Nievergelt, and N. Deo, *Combinatorial Algorithms*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1977.

Received October 1, 1990; accepted for publication November 1, 1991

Ching-Tien Ho *IBM Research Division, Almaden Research Center, 650 Harry Road, San Jose, California 95120 (HO at ALMADEN, ho@almaden.ibm.com).* Dr. Ho received a B.S. degree in electrical engineering from the National Taiwan University in 1979 and M.S., M.Phil., and Ph.D. degrees in computer science from Yale University in 1985, 1986, and 1990, respectively. Since August 1989, he has been a Research Staff Member in the Foundations of Massively Parallel Computing group at the IBM Almaden Research Center, San Jose, California. His primary research interests include communication issues for interconnection networks, algorithms for collective communications, graph embeddings, fault tolerance, and parallel algorithms and architectures. Dr. Ho is a co-recipient of the 1986 "Outstanding Paper Award" of the International Conference on Parallel Processing. He has received an Outstanding Innovation Award and three Plateau Invention Achievement Awards from IBM. He is a member of the Association for Computing Machinery and the IEEE Computer Society.

S. Lennart Johnsson *Department of Computer Science, Harvard University, Cambridge, Massachusetts; also with Thinking Machines Corporation, 245 First Street, Cambridge, Massachusetts 02142 (johnsson@cs.harvard.edu).* Dr. Johnsson received the M.S. and Ph.D. degrees from Chalmers Institute of Technology, Gothenburg, Sweden, in 1967 and 1970, respectively. From 1970 to 1979 he was affiliated with the Central Research and Development Laboratories of ASEA AB, Sweden. From 1979 to 1983 he was a Senior Research Associate in computer science at the California Institute of Technology. In 1983 he joined the faculty of Yale University, where he introduced courses on parallel algorithms and architectures, and continued his research on architectures, algorithms, and software for high-performance parallel computers for the computational sciences. Since 1986 Dr. Johnsson has been Director of Computational Sciences at Thinking Machines Corporation, where he is leading the development of efficient communication routines, and the development of the Connection Machine® Scientific Software Library, CMSSL. Since 1990, Dr. Johnsson has been Gordon McKay Professor of the Practice of Computer Science at Harvard University, where he introduced education in scientific computation on parallel scalable architectures. He is an editor of several scientific journals and the author or co-author of more than ninety articles and conference papers, and numerous technical reports. He is a co-recipient of the 1986 Outstanding Paper Award of the International Conference on Parallel Processing. Dr. Johnsson is a Board Member of the Computing Research Association and serves on the USRA Science Council for CESDIS. He has served on the program and organizing committees for several conferences on parallel computing.