# Yale University
# Department of Computer Science

**Embedding Hyper-Pyramids into Hypercubes**

Ching-Tien Ho and S. Lennart Johnsson

YALEU/DCS/TR-667
December 1988

This work has in part been supported by the Office of Naval Research under contract N00014-86-K-0310. Approved for public release: distribution is unlimited.

## Embedding Hyper-Pyramids into Hypercubes

Ching-Tien Ho and S. Lennart Johnsson[1]
Department of Computer Science
Yale University
New Haven, CT 06520
Ho@cs.yale.edu, Johnsson@cs.yale.edu, Johnsson@think.com

**Abstract.** A $\hat{P}(k,d)$ hyper-pyramid is a level structure of $k$ Boolean cubes where the cube at level $i$ is of dimension $id$, and a node at level $i-1$ connects to every node in a $d$ dimensional Boolean subcube at level $i$, except for the leaf level $k$. Hyper-pyramids contain pyramids as proper subgraphs. We show that a $\hat{P}(k,d)$ hyper-pyramid can be embedded in a Boolean cube with minimal expansion and dilation 2. The congestion is bounded from above by $\frac{2^{d+1}}{d+1}$ and from below by $1+\lceil\frac{2^d-d}{kd+1}\rceil$. For $\hat{P}(k,2)$ hyper-pyramids we present a dilation 2 and congestion 2 embedding. In addition to expansion, dilation, and congestion we also characterize the embedding with the active-degree, and the node-load. The former property gives the maximum number of cube edges being used at any node, and the latter property measures the maximum number of messages a cube node needs to handle. The active degree for the embeddings is equal to the number of cube edges per node, i.e., $kd+1$, and the node-load is bounded from above by $O(2^d)+O(kd)$ with a congestion of $O(\frac{2^d}{d})$. For the $\hat{P}(k,2)$ hyper-pyramid embedding we present, the node-load is $2k+5$.

We also present embeddings of a $\hat{P}(k,d)$ hyper-pyramid together with $2^d-2$ $\hat{P}(k,d)$ hyper-pyramids such that only one cube node is unused. The dilation of the embedding is $d+1$ with a congestion of $O(2^d)$. An alternate embedding with dilation $2d$ and congestion $O(\frac{2^d}{d})$ is also presented. The active-degree is $kd+1$ for both embeddings. The node-load is $O(d2^d)+O(kd)$ for the former and $O(2^d)+O(kd)$ for the latter embedding. Specialized to hyper-pyramids $\hat{P}(k,2)$ we present two embeddings: one with dilation 3, congestion 3 and a node-load of $3k+5$; the other with dilation 4, congestion 5 and a node-load of $2k+9$. As a corollary a complete $n$-ary tree can be embedded in a Boolean cube with dilation $\max(2,\lceil\log_2 n\rceil)$ and expansion $2^{k\lceil\log_2 n\rceil+1}/\frac{n^{k+1}-1}{n-1}$.

**Key words.** embedding, hypercube, hyper-pyramid, expansion, dilation, congestion, active-degree, node-load

# 1 Introduction

Processor utilization and communication time are two important considerations in selecting data structures and algorithms for architectures assembled out of a large number of parts. Communication is one of the most expensive resources in such an architecture, and its efficient

[1]Also with Dept. of Electrical Engineering, Yale Univ., and Thinking Machines Corp., Cambridge, MA 02142.

utilization is imperative. In studying the efficient utilization of the communication system the communication needs of the computations are modeled by a graph, which is usually referred to as the *guest graph* [29]. This graph discloses the interaction between the data elements of the computation. Similarly, the topology of the ensemble architecture is captured by a graph, the *host graph.* Each vertex represents a processor with local storage and each edge a communication link between processors. The guest graph is embedded in the host graph for the execution.

The embedding function $f$ maps each vertex in the guest graph $G$ into a unique vertex in the host graph $H$. Let $\mathcal{V}(G)$ and $\mathcal{E}(G)$ denote the node sets and the edge sets of a graph $G$. $|S|$ denotes the cardinality of a set $S$. The *expansion* of the mapping is $\frac{|\mathcal{V}(H)|}{|\mathcal{V}(G)|}$. Each edge $e_G = (i,j) \in \mathcal{E}(G)$ is mapped into a path in $H$, $path_H(e_G)$. The *dilation* of the mapping is the maximum length of $path_H(e_G)$ for all $e_G \in \mathcal{E}(G)$. The expansion is a measure of processor utilization. The slow down of nearest-neighbor communication in the original graph by edges being "stretched" into paths of length greater than one is a function of the dilation and *congestion*, the maximum number of guest graph edges sharing an edge in the host graph. From an architectural point of view it is also of interest to know how many edges of a node in the host graph are being used by the embedding, and the total number of messages being serviced by a node when every node sends a message to the neighboring nodes as defined by the guest graph. The first quantity is measured by the *active-degree* and the latter by the *node-load.* Related to the embedding of pyramids is the embedding of meshes and trees. Embedding of meshes into hypercubes has been studied in [20,17,7,15,12,6]. Embedding of trees into hypercubes has been studied in [13,22,18,17,35,3,2,10,19,16,34,26].

Several parallel algorithms that naturally lend themselves to a pyramid topology are discussed, for instance, in [32,27,9,5,33]. Multigrid algorithms for partial differential equations [8] and algorithms for image processing [5] are specific examples. The embedding of pyramids into hypercubes was first studied by Stout [31]. He proved that there exists a constant dilation embedding of an $M$ node pyramid in an $N$ node Boolean cube with $N \ll M$, if $\approx \frac{M}{N}$ pyramid nodes are mapped to every cube node. Stout also showed that for a one-to-one mapping minimal expansion and dilation 2 is possible. Lai and White [23] give 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. Independently, Leighton et al. [25] have obtained a similar result. We also generalize previous embeddings to minimal expansion and dilation $d$ embeddings of *hyper-pyramids*, graphs where each non-leaf node has $2^d$ children and the nodes at the same level form a hypercube (instead of a mesh).

Lai and White [24] also gave an algorithm for embedding a pyramid and two smaller pyramids (of approximately quarter size each) into a hypercube with expansion $\approx 1$ dilation 3 and congestion 6. We improve the result to expansion $\approx 1$, dilation 3, and congestion 3. The result is generalized to the embedding of one hyper-pyramid with minimal expansion, and the embedding of $2^d - 2$ smaller hyper-pyramids in the same Boolean cube with a total expansion $\approx 1$, and a dilation of $d+1$. We also consider active-degree and node-load for all the embeddings presented, which are summarized in Table 3 in Section 4.

In the next section we introduce the notation used in the paper, define pyramids and hyper-pyramids, and give some of their properties. Section 3 contains the main results, which are summarized in Section 4.

## 2 Preliminaries

Let $Hamming(x, y)$ be the Hamming distance between $x$ and $y$ and $||x||$ be the number of 1-bits in the binary representation of $x$, i.e., $||x|| = Hamming(x, 0)$. Also, let $0^m$ be a string of $m$ 0-bits, and $1^m$ a string of $m$ 1-bits. Let $j_m$ be the $m$th bit of the binary representation of $j$ with the least significant bit being the $0th$ bit. Let $\mathcal{A}_G(i)$ be the set of nodes adjacent to node $i$: $\mathcal{A}_G(i) = \{j | (i, j) \in \mathcal{E}(G)\}$. Let $x^{\{m\}} = x \oplus 2^m$.

**Definition 1** An embedding $f$ of a *guest* graph, $G$, into a *host* graph, $H$, is a one-to-one mapping from $\mathcal{V}(G)$ to $\mathcal{V}(H)$. The *expansion* of the embedding $f$ is

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

In order to consider *dilation* and *congestion*, we specify the path from $f(i)$ to $f(j)$ in $H$ for every edge $e_G = (i, j) \in \mathcal{E}(G)$. Let $path_H(e_G) = p_0, p_1, \ldots, p_k$, where $p_m \in \mathcal{V}(H)$, for all $0 \leq m \leq k$, $p_0 = f(i)$ and $p_k = f(j)$. Moreover, let $\mathcal{E}(path_H(e_G)) = \{(p_m, p_{m+1}) | 0 \leq m < k\}$, i.e., the set of edges along the path.

**Definition 2** The dilation of an edge $e_G \in \mathcal{E}(G)$ is the length of the path $path_H(e_G)$:

$$dil_f(e_G) = |\mathcal{E}(path_H(e_G))|.$$

The *dilation* of the embedding $f$ is

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

We will 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 $e_H \in \mathcal{E}(H)$, $cong_f(e_H)$, is the number of edges in $G$ with images including $e_H$,

$$cong_f(e_H) = \sum_{\forall e_G \in \mathcal{E}(G)} |\{e_H\} \cap \mathcal{E}(path_H(e_G))|.$$

The *congestion* of the mapping $f$ is

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

*Congestion* is sometimes referred to as *load-factor* [2].

**Definition 4** The *active-degree* of a node $i$, $\alpha_i$, is the number of edges of host graph node $i$ being part of any $path_H$.

$$\alpha_i = \sum_{\forall j \in \mathcal{A}_H(i)} |\{(i, j)\} \cap (\cup \mathcal{E}(path_H(e_G)), \forall e_G \in \mathcal{E}(G))|.$$

The *active-degree* of the mapping $f$ is

$$\textit{active-degree}_f = \max_{\forall i \in \mathcal{V}(H)} \alpha_i.$$

The *active-degree* is a measure of the number of ports that need to be serviced concurrently in case the communication is pipelined. The *node-load* measures the total load on any node, which is of particular importance in case a node can service only one port at a time. The number of messages a node has to service is the sum of the degree of the pyramid node mapped to it and twice the number of paths going through it (one send, one receive operation).

**Definition 5** The *node-load* of node $i$, $\beta_i$, is the number of messages that node $i$ needs to service

$$\beta_i = \sum_{\forall j \in \mathcal{A}_H(i)} cong_f((i,j)).$$

The *node-load* of the mapping $f$ is

$$\textit{node-load}_f = \max_{i \in \mathcal{V}(H)} \beta_i.$$

**Definition 6** A $l_1 \times l_2$ mesh $M(l_1, l_2)$ is a graph with vertex set

$$\mathcal{V}(M(l_1,l_2)) = \{(x_1,x_2) | 0 \leq x_1 < l_1, 0 \leq x_2 < l_2\}$$

and edge set

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

**Definition 7** A $k$-level pyramid $P(k, l_1, l_2)$ is a graph with vertex set

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

and edge set

$$\begin{aligned}\mathcal{E}(P(k,l_1,l_2)) &= \bigcup_{i=0}^{k} \{((i,x_1,x_2),(i,x_1',x_2')) | ((x_1,x_2),(x_1',x_2')) \in \mathcal{E}(M(l_1^i,l_2^i))\} \bigcup \\ &\quad \bigcup_{i=1}^{k} \{((i,x_1,x_2),(i-1,\left\lfloor \frac{x_1}{l_1} \right\rfloor, \left\lfloor \frac{x_2}{l_2} \right\rfloor)) | (x_1,x_2) \in \mathcal{V}(M(l_1^i,l_2^i))\}.\end{aligned}$$

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 the same as $P(k, l_2, l_1)$. Figure 1 shows the topology of a $P(2,2,2)$ pyramid. It can be viewed as a complete quad-tree with nodes at the same level being connected as a mesh.

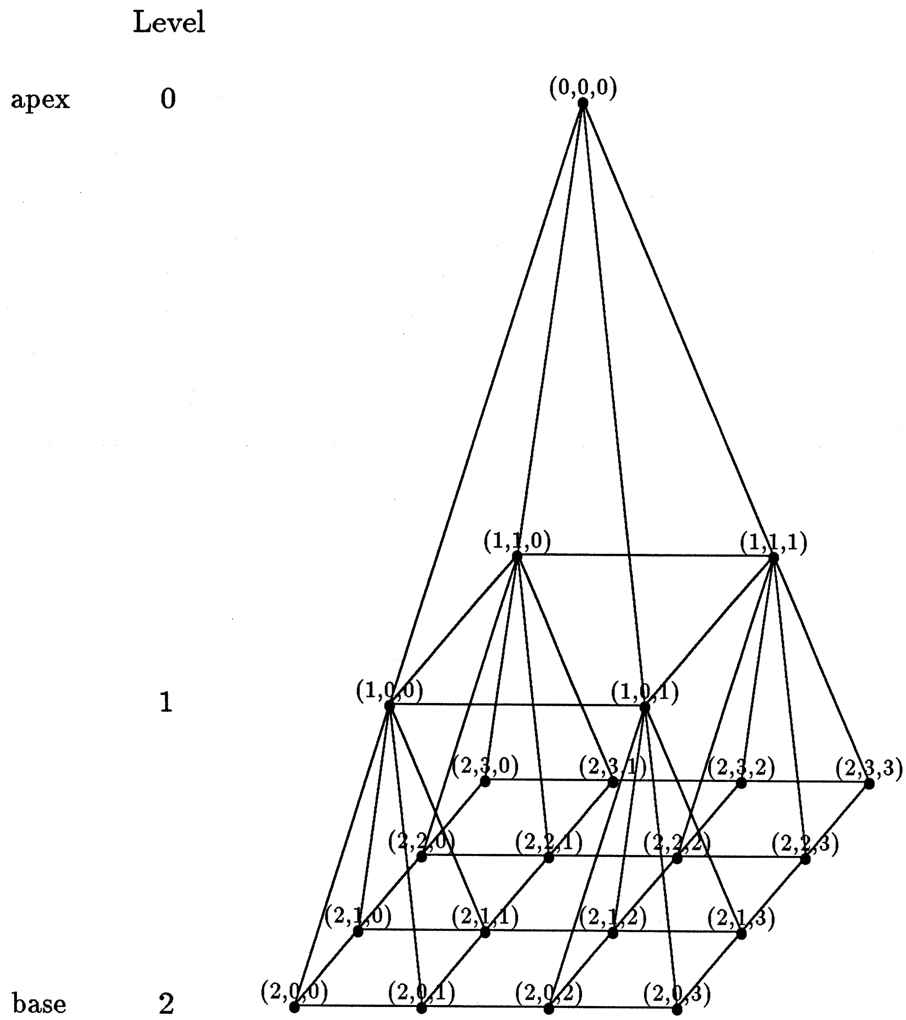



Figure 1: The topology of a $P(2,2,2)$.

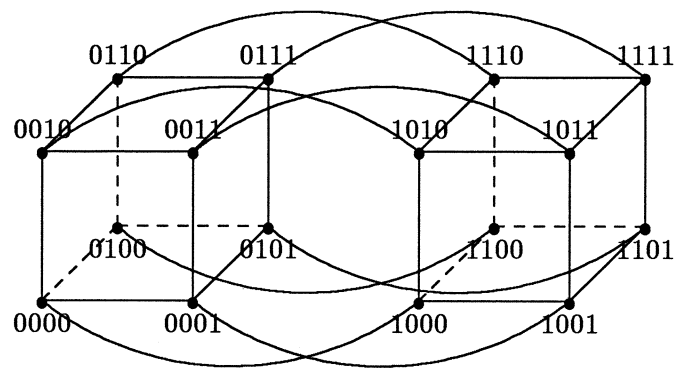



Figure 2: The $H_4$ with corresponding node addresses used in this paper.

The number of vertices in a pyramid is

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

and the number of edges is

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

Denote a $k$-dimensional Boolean cube by $H_k$. Figure 2 shows an $H_4$. The addresses of cube nodes in subsequent figures are omitted, but determined in the same way. For clarity we omit edges in certain high dimensions in subsequent figures.

**Definition 8** A *k-level hyper-pyramid* $\hat{P}(k, d)$ of *degree d* is defined recursively as follows. $\hat{P}(0, d)$ is a root node. A $\hat{P}(k, d)$ hyper-pyramid is constructed out of $2^d$ $\hat{P}(k-1, d)$ hyper-pyramids by interconnecting corresponding nodes in each of these hyper-pyramids as $d$-dimensional Boolean cubes, and connecting a new node to every root of the $\hat{P}(k-1, d)$ hyper-pyramids.

**Lemma 1** *A $\hat{P}(k, d)$ hyper-pyramid contains a $P(k, 2^i, 2^{d-i})$ pyramid, $0 \leq i \leq d$, as a subgraph.*

As a corollary, a $\hat{P}(k, 2)$ hyper-pyramid contains a $P(k, 2, 2)$ pyramid as a subgraph. In the following we will only consider the embedding of hyper-pyramids into Boolean cubes.

We will use definition 8 in specifying embedding functions, $f$, and prove their properties with respect to dilation, congestion, active-degree, and node-load. Hyper-pyramids can also be defined recursively by adding an $H_{kd}$ cube to a $\hat{P}(k-1, d)$ hyper-pyramid. The hyper-pyramid $\hat{P}(k, d)$ is obtained by connecting each node in the $H_{kd}$ to a (parent) node in the base of the $\hat{P}(k-1, d)$ hyper-pyramid. Such a definition emphasizes the fact that hyper-pyramids can be viewed as a sequence of cubes of linearly increasing dimensions with a tree structure connecting them.

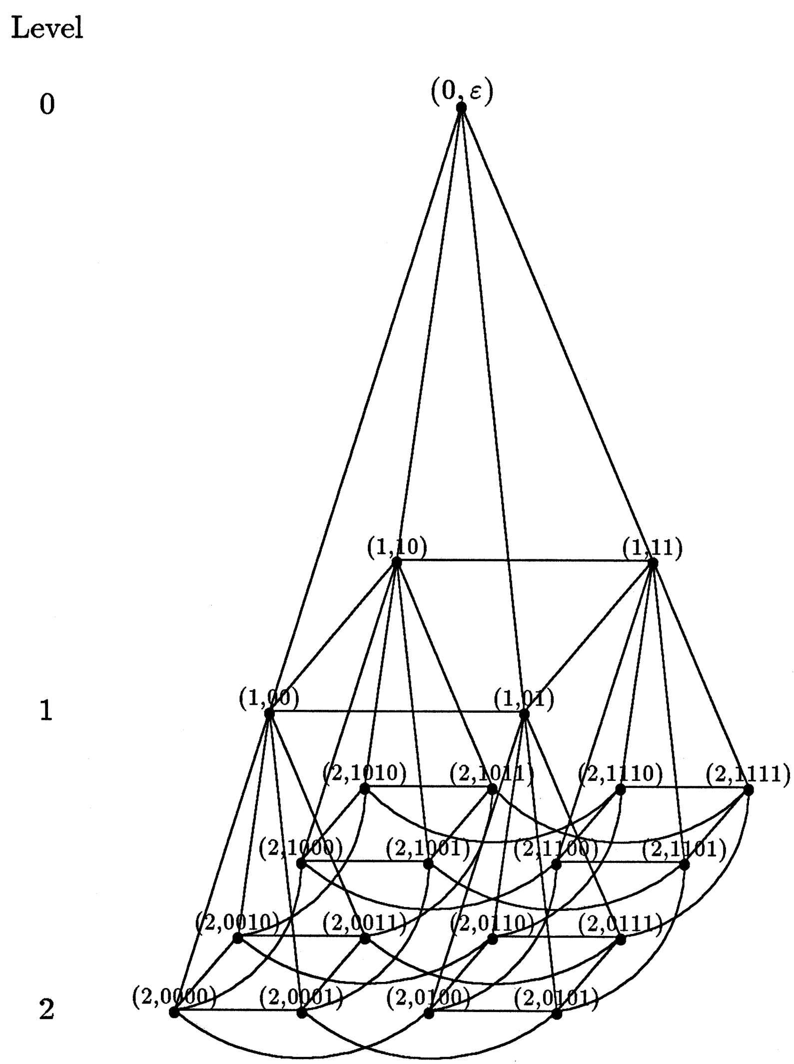



Figure 3: The topology of a $\hat{P}(2,2)$ hyper-pyramid.

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

Figure 3 shows the topology of a $\hat{P}(2,2)$ hyper-pyramid. Note that $id$ bits are used for the second argument of the node addresses at level $i$. The second argument of a root node is a null string, which is represented by $\varepsilon$. Figure 4 gives another view of the same hyper-pyramid.

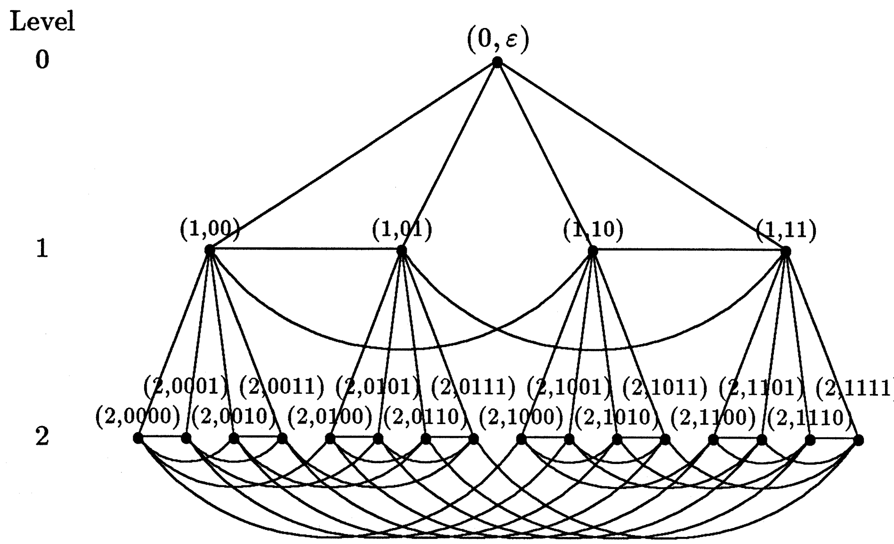



Figure 4: Another view of the topology of a $\hat{P}(2,2)$ hyper-pyramid.

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

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

and the number of edges is

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

In the formula for the number of edges the first term accounts for the edges at each level and the second term accounts for the edges between levels. From Figures 1 and 3 it is clear that a $P(2,2,2)$ pyramid with end-around edges added to the mesh at level 2 is topologically equivalent to a $\hat{P}(2,2)$ hyper-pyramid. This is because a $4 \times 4$ torus is topologically equivalent to an $H_4$ Boolean cube (and, in general, a $d$-dimensional torus with width 4 is topologically equivalent to an $H_{2d}$ cube).

**Lemma 2** *A lower bound for the dilation of any embedding of a $\hat{P}(k,d)$ hyper-pyramid into a smallest Boolean cube $H_n$ (i.e., $|\mathcal{V}(H_{n-1})| \leq |\mathcal{V}(\hat{P}(k,d))| < |\mathcal{V}(H_n)|$) is $\frac{d}{2}$.*

**Proof:** The diameter of a $\hat{P}(k,d)$ hyper-pyramid is $2k$. The smallest cube $H_n$ which is large enough to hold a $\hat{P}(k,d)$ hyper-pyramid has $n = kd+1$ dimensions. Since the hyper-pyramid contains more than $2^{n-1}$ nodes, there exist two hyper-pyramid nodes that are mapped to cube nodes at a distance of at least $n-1$ in the $H_n$ cube. Consider any shortest path between these two hyper-pyramid nodes. Let the length of the path be $\ell$. Clearly, $\ell \leq 2k$. Each edge on the path will be stretched in the embedding such that all $\ell$ edges together are stretched into the path of length $\geq n-1$ in the $H_n$ cube. So, at least one of this $\ell$ edges is stretched into a path of length

$$\geq \frac{n-1}{\ell} \geq \frac{n-1}{2k} = \frac{d}{2}. \quad \blacksquare$$

Similarly, a lower bound of the dilation for embedding a $P(k, l_1, l_2)$ pyramid into a $H_{1+k\log_2 l_1 l_2}$ cube is $\frac{\log_2 l_1 l_2}{2}$. The following four lemmas will be used later for proofs of congestion and node-load.

**Lemma 3** *A $2^n$ node flat tree (i.e., a root with $2^n - 1$ children) can be embedded in an $H_n$ cube with congestion $\leq \frac{2^{n+1}}{n+2}$.*

**Proof:** Embed the flat tree into the Spanning Balanced $n$-Tree of an $n$-cube [19,16]. Then each subtree of the root has at most $\frac{2^{n+1}}{n+2}$ nodes, and the congestion is dominated by the out-going edges of the root, which in turn is equal to the number of nodes in the subtree. $\blacksquare$

An asymptotically better upper bound on the number of nodes in a subtree of the Spanning Balanced $n$-Tree can be derived from [16] as $(1+\frac{4}{n})\frac{2^n-2}{n}+1$.

**Lemma 4** *A $2^n - 1$ node flat tree can be embedded in an $H_n$ cube such that the congestion is at most $\frac{2^{n+1}}{n+1}$ and the node-load of the unused cube node is zero (i.e., no path passes through the unused cube node).*

**Proof:** Let $j$ be one of the dimensions in which the corresponding bits of the root and the unused node addresses differ. Partition the $H_n$ cube into two subcubes with respect to dimension $j$. Call the subcube which contains the root of the flat tree subcube 0, and the other subcube 1. Define a balanced tree in subcube 0. Then, extend the tree by connecting each node in subcube 0 to its corresponding node in subcube 1 except that the unused node is not included. The congestion of the embedding is dominated by the out-going edges of the root, i.e., twice the number of nodes in the maximum subtree of the balanced tree in subcube 0. This number is $\leq 2 \times \frac{2^n}{n+1} = \frac{2^{n+1}}{n+1}$. $\blacksquare$

**Definition 9** A *translation of a node* $x$ with respect to a node $s$ is a bit-wise exclusive-or of the two node addresses $x$ and $s$, i.e., $Tr(x,s) = x \oplus s$. A *translation of a graph* $G(\mathcal{V},\mathcal{E})$ with respect to a node $s$ is a translation of all the vertices of the graph $G$ with respect to the node $s$, i.e., $Tr(G(\mathcal{V},\mathcal{E}),s) = G(Tr(\mathcal{V},s), Tr(\mathcal{E},s))$ where $Tr(\mathcal{V},s) = \{Tr(x,s) | \forall x \in \mathcal{V}\}$ and $Tr(\mathcal{E},s) = \{(Tr(x,s), Tr(y,s)) | \forall (x,y) \in \mathcal{E}\}$.

**Lemma 5** *A $2^n$ node complete graph, with all edges duplicated, can be embedded into an $H_n$ Boolean cube with congestion $\leq \frac{n2^{n+1}}{n+2} + n$.*

**Proof:** By duplicating all edges in the complete graph, one can decompose the edges of the complete graph into sets $E_0, E_1, \ldots, E_{2^n-1}$ such that the graph $G_i = (V, E_i)$ forms a $2^n$ node flat tree rooted at node $i$, where $V$ is the node set of the complete graph. Embed the flat tree $G_i$ into the Spanning Balanced $n$-Tree (SBnT) [19] rooted at node $i$ in an $H_n$ cube. In order to consider the congestion, we modify the SBnT such that every edge which connects a node to its child node $x$ is replaced by $g_x$ edges where $g_x$ is the number of nodes in the subtree rooted at node $x$ (including the node $x$). The congestion for the embedding of these $2^n$ flat trees in an $H_n$ cube is equal to the maximum number of edges of the $2^n$ modified SBnT's that share a cube edge. Since the SBnT's are distinct translations of each other (with respect to its own root address), cube edges of the same dimension are evenly used. The edge congestions of all cube edges in the same dimension, say dimension $y$, are the same, which is equal to twice the number of tree edges in dimension $y$ of a modified SBnT. Furthermore, from the property that subtrees of the SBnT can be obtained through address rotation of each other [14], the number of tree edges in a given dimension of a modified SBnT can be shown to be bounded from above by the number of tree edges in the largest subtree of a modified SBnT. Also, from a correspondence property between nodes at level $\ell$ and level $n - \ell$ within subtrees [14], the number of tree edges in the largest subtree of a modified SBnT is: (the number of nodes in the largest subtree of the SBnT plus one) times $\frac{n}{2}$. The congestion is bounded from above by

$$2\left(\frac{2^{n+1}}{n+2} + 1\right)\frac{n}{2} = \frac{n2^{n+1}}{n+2} + n. \qquad \blacksquare$$

**Lemma 6** *A $2^n - 1$ node complete graph can be embedded into an $H_n$ cube such that the congestion is $O(2^n)$ and the node-load of the unused cube node is zero (i.e., no path passes through the unused cube node).*

The proof follows that of Lemma 5, but each SBnT is modified to exclude the unused node as described in the proof of Lemma 4. Clearly, the order of the congestion is the same as that in Lemma 5.

## 3 Embedding hyper-pyramids into hypercubes

The main results of this paper are:

1. A $\hat{P}(k,d)$ hyper-pyramid, $d \geq 2$, can be embedded into an $H_{kd+1}$ Boolean cube with expansion $< 2$ and dilation $d$. The congestion is bounded from below by $1 + \lceil \frac{2^d - d}{kd+1} \rceil$ and from above by $\lceil \frac{2^{d+1}}{d+1} \rceil$. The active-degree of the mapping is $kd + 1$ and the node-load is bounded from above by $(1 + \frac{4}{d+1})2^d + (k-1)d$.

2. A $\hat{P}(k,2)$ hyper-pyramid can be embedded into an $H_{2k+1}$ Boolean cube with expansion $<$ 2, dilation 2, and congestion 2. The active-degree of the embedding is $2k + 1$, and the node-load is $2k + 5$.

3. A $\hat{P}(k,d)$ hyper-pyramid together with $(2^d-2)$ $\hat{P}(k-1,d)$ hyper-pyramids, $d \geq 2$, can be embedded into an $H_{kd+1}$ Boolean cube with expansion $\approx 1$ (only one cube node is not used) and dilation $d+1$. The congestion is at most $O(2^d)$ and the node-load is at most $O(d2^d)+O(kd)$. However, with a dilation of $2d$, the congestion is reduced to at most $\lceil \frac{2^d}{d} \rceil + \frac{2^{d+1}}{d+2} + 1$ and the node-load to at most $(3+\frac{4}{d+2})2^d + (k-3)d$, $k \geq 3$. Both embeddings have active-degree $kd+1$.
4. A $\hat{P}(k,2)$ hyper-pyramid and two $\hat{P}(k-1,2)$ hyper-pyramids can be embedded into an $H_{2k+1}$ Boolean cube with expansion $\approx 1$, dilation 3, and congestion 3. The active-degree is $2d+1$ and the node-load is $3k+5$. Alternatively, with dilation 4 and congestion 5, the node-load can be $2k+9$.

The bounds are summarized in Table 3 in Section 4. (2) is a special case of (1), and (4) is a special case of (3). The expansion of (1) is $\frac{2(2^d-1)}{2^d-2^{-k}}$, which is bounded from above by $\frac{2^{d+1}}{2^d+1}$ and from below by $\frac{2^d-1}{2^{d-1}}$ for $k \geq 1$. For $d=2$, it is between $\frac{8}{5}$ and $\frac{3}{2}$.

**Definition 10** A *two-rooted hyper-pyramid* $\tilde{P}(k,d)$ is a $\hat{P}(k,d)$ hyper-pyramid with an additional root node and additional edges between it and all nodes at level 1. The two roots are denoted $a(0,\varepsilon)$ and $a(0',\varepsilon)$, respectively.

Since the two roots are symmetrical, either one can serve as the root. One of the roots will be a node at level one after the induction step. This root is the *real root.* The other of the two roots will either serve as one of the two new roots, or become unused. This root is the *spare root.* There is no edge between the two roots according to the definition, but the embedding functions presented below always map the two roots to adjacent cube nodes. The idea of using two roots for the recursive construction of tree structures has been used before, for instance, by Bhatt and Leiserson [4] in constructing a complete binary tree out of chips containing smaller trees, and by Bhatt and Ipsen [3] in embedding complete binary trees into a Boolean cube.

## 3.1 Embedding a $\hat{P}(k,d)$ hyper-pyramid in a Boolean cube with minimal expansion and dilation $d$

### 3.1.1 Dilation

**Theorem 1** *A $\hat{P}(k,d)$ hyper-pyramid, $d \geq 2$, can be embedded into an $H_{kd+1}$ Boolean cube with dilation d.*

**Proof:** Instead of considering the embedding of a $\hat{P}(k,d)$ hyper-pyramid we consider the embedding of the corresponding two-rooted hyper-pyramid $\tilde{P}(k,d)$. The dilation for the two-rooted hyper-pyramid is an upper bound on the dilation for the hyper-pyramid with a single root. Let $f_k$ be the function that maps a two-rooted hyper-pyramid $\tilde{P}(k,d)$ into an $H_{kd+1}$ cube with dilation $d$. We will define $f_k$ (for a fixed $d$) by a recursive construction on $k$ and prove the theorem by induction. The induction hypothesis is that for $k \leq n$, a two-rooted hyper-pyramid $\tilde{P}(k,d)$

can be embedded into an $H_{kd+1}$ cube with dilation $d$ and the two roots mapped to adjacent cube nodes.

**Basis:** For $k = 0$, the two-rooted hyper-pyramid $\tilde{P}(0,d)$, which has only two roots, are mapped to adjacent cube nodes:

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

**Induction:** Assume that there exists an embedding function $f_n$ which satisfies the induction hypothesis. In order to embed a two-rooted hyper-pyramid $\tilde{P}(n+1,d)$ into an $H_{(n+1)d+1}$ cube, we consider the cube $H_{(n+1)d+1}$ as composed of $2^d$ copies of $H_{nd+1}$ cubes, labeled $0, 1, \ldots, 2^d - 1$. Apply $f_n$ to each $H_{nd+1}$ cube. We use a superscript to distinguish nodes of different two-rooted hyper-pyramids $\tilde{P}(n,d)$ mapped to distinct $H_{nd+1}$ cubes. The following rules define the embedding function $f_{n+1}$ in terms of $f_n$ for each cube.

$$\begin{aligned}
f_{n+1}(a(0,\varepsilon)) &= f_n(a^0(0,\varepsilon)), \\
f_{n+1}(a(0',\varepsilon)) &= f_n(a^{2^{d-1}}(0,\varepsilon)), \\
f_{n+1}(a(1,\ell)) &= \begin{cases} f_n(a^\ell(0',\varepsilon)), & \ell = 0 \text{ or } 2^{d-1}, \\ f_n(a^\ell(0,\varepsilon)), & \text{otherwise}, \end{cases} \\
f_{n+1}(a(i,\ell||j)) &= f_n(a^\ell(i,j)), \qquad i > 1.
\end{aligned}$$

The first two equations define the two new roots. (The two roots can be chosen from the spare roots of any two adjacent cubes. We choose cubes 0 and $2^{d-1}$ such that the two roots are mapped to cube addresses $(00\ldots0)$ and $(10\ldots0)$, respectively.) The third equation defines nodes at level 1. The last equation defines nodes at lower levels, where $\ell$ is a string of length $d$. Figures 5 and 6 show the induction for $d = 2$ and $d = 3$, respectively. For clarity, only the two roots of each cube are shown. In the figures, $\phi$ denotes unused cube nodes (after induction), and "$\rightarrow$" reads "becomes". Note that the root $a(0',\varepsilon)$ is used in cubes 0 and $2^{d-1}$. Figures 7 and 8 show the embeddings for $\tilde{P}(1,2)$ and $\tilde{P}(1,3)$, respectively. For all $0 \le j < 2^d$ and $0 \le m < d$, we have the following properties:

1. $\mathit{Hamming}(f_{n+1}(a(0,\varepsilon)), f_{n+1}(a(1,j))) \le d$:
   $\mathit{Hamming}(f_{n+1}(a(0,\varepsilon)), f_{n+1}(a(1,j))) =$
   $$\begin{cases} \mathit{Hamming}(f_n(a^0(0,\varepsilon)), f_n(a^j(0,\varepsilon))) = ||j|| < d, & \text{if } j \ne 0 \text{ and } j \ne 2^{d-1}, \\ \mathit{Hamming}(f_n(a^0(0,\varepsilon)), f_n(a^0(0',\varepsilon))) = 1, & \text{if } j = 0, \\ \mathit{Hamming}(f_n(a^0(0,\varepsilon)), f_n(a^{2^{d-1}}(0',\varepsilon))) = 2, & \text{if } j = 2^{d-1}. \end{cases}$$
2. $\mathit{Hamming}(f_{n+1}(a(0',\varepsilon)), f_{n+1}(a(1,j))) \le d$: The proof follows that of 1.
3. $\mathit{Hamming}(f_{n+1}(a(1,j)), f_{n+1}(a(1,j^{\{m\}}))) \le 2$: The distance is 1 except if $m \ne d-1$ *and* $j = 0$ or $2^{d-1}$ for which the distance is 2.
4. $\mathit{Hamming}(f_{n+1}(a(0,\varepsilon)), f_{n+1}(a(0',\varepsilon))) = 1$.
5. The Hamming distance between corresponding nodes of adjacent cubes is 1.

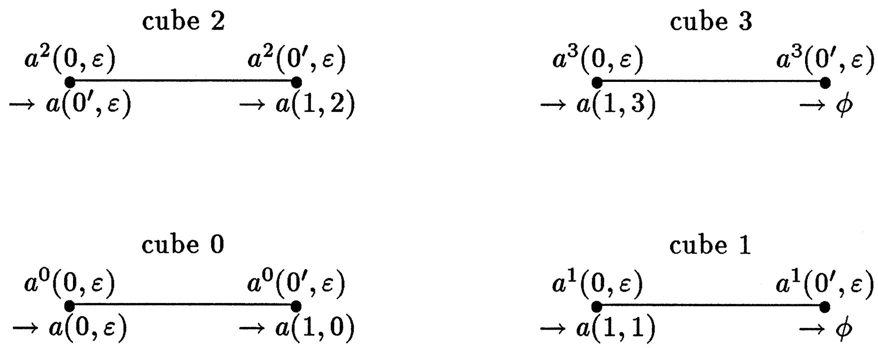



Figure 5: Forming a two-rooted hyper-pyramid $\tilde{P}(n+1,2)$ out of 4 copies of $\tilde{P}(n,2)$ hyper-pyramids.

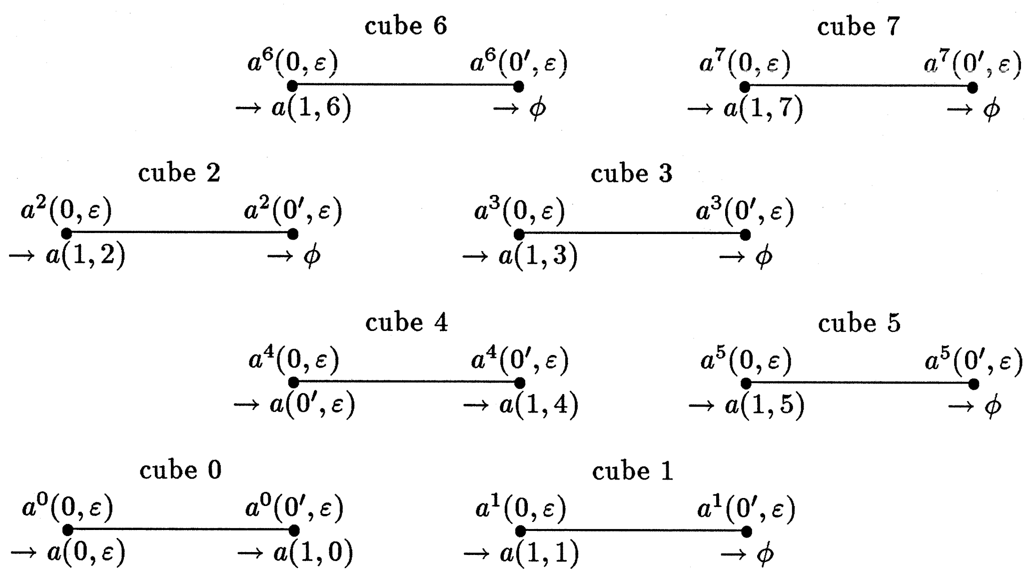



Figure 6: Forming a two-rooted hyper-pyramid $\tilde{P}(n+1,3)$ out of 8 copies of $\tilde{P}(n,3)$ hyper-pyramids.

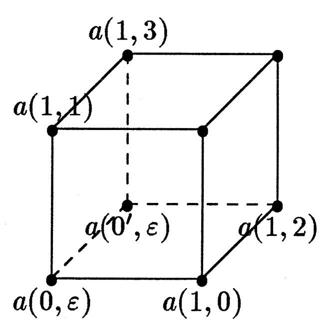



Figure 7: A two-rooted $\tilde{P}(1,2)$ hyper-pyramid embedded in an $H_3$ cube with dilation 2.

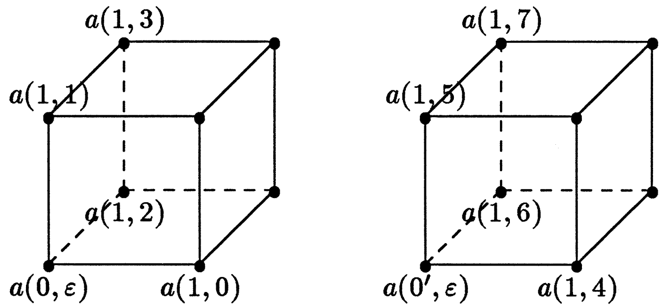



Figure 8: A two-rooted $\tilde{P}(1,3)$ hyper-pyramid embedded in an $H_4$ cube with dilation 3.

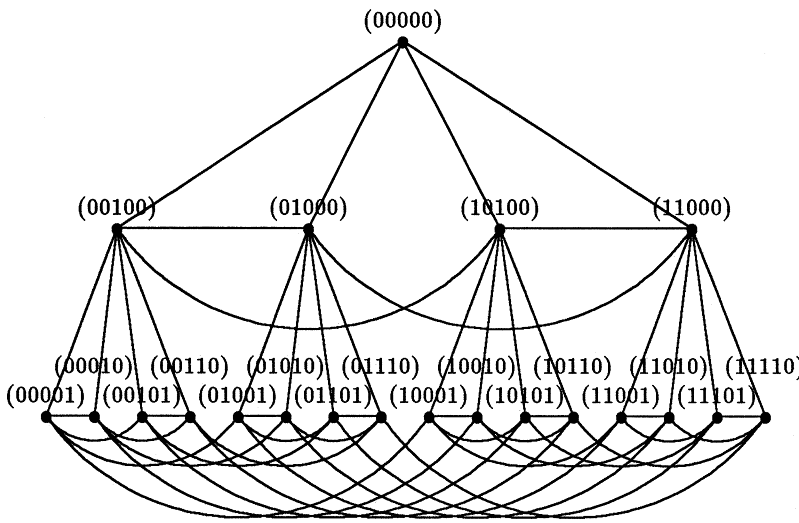



Figure 9: The cube addresses of the nodes of an embedded hyper-pyramid $\hat{P}(2,2)$.

6. The dilation of an edge in $\tilde{P}(n,d)$ is preserved.

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

$$f_k(a(i,j)) = \begin{cases} (0^{kd+1}), & i = 0, \\ (10^{kd}), & i = 0', \\ (j||x0^{(k-i)d}), & 1 \leq i \leq k, \end{cases}$$

where $x = 1$, if $(j_{d-2}j_{d-3}\cdots j_0) = 0$; and $x = 0$, otherwise.

Figure 9 shows the cube addresses of the nodes of the hyper-pyramid $\hat{P}(2,2)$.

The expansion of the embedding function $f_k$ is less than 2 (except for $k = 0$).

### 3.1.2 Congestion

Next we derive upper and lower bounds for the congestion.

**Lemma 7** *An upper bound of the congestion for embedding a $\hat{P}(k,d)$ hyper-pyramid in an $H_{kd+1}$ Boolean cube is $\frac{2^{d+1}}{d+1}$.*

**Proof:** The proof can be done by induction based on the following arguments. The maximum edge congestion is caused by the hyper-pyramid edges between the root node and its $2^d$ children. Among the $2^d$ children, $2^d-2$ of them are in an $H_d$ cube. The other two children are neighbors of the two roots, but not contained in the $H_d$ cube. The two roots are in the same $H_d$ cube as the $2^d-2$ children. By Lemma 4, the congestion caused by the edges between the real root and its children in the $H_d$ cube is bounded from above by $\frac{2^{d+1}}{d+1}$. (We perform the hyper-pyramid embedding such that the two roots are adjacent, and avoid routing the edges from the real root through the spare root, in order to minimize the node-load.) We route the $d-1$ length-two paths from node $a(1,0)$ or $a(1,2^{d-1})$ to its $d-1$ neighbors through an unused cube node. The path between nodes $a(0,\varepsilon)$ and $a(1,2^{d-1})$ is routed through node $a(1,0)$. Note that the congestion of the edges in the $H_d$ cube does not increase for the next induction step. ∎

**Lemma 8** *A lower bound of the congestion for embedding a $\hat{P}(k,d)$ hyper-pyramid in an $H_{kd+1}$ Boolean cube is $1+\lceil\frac{2^d-d}{kd+1}\rceil$.*

**Proof:** The nodes at level $k-1$ of a hyper-pyramid $\hat{P}(k,d)$ have degree $1+(k-1)d+2^d$. The degree of an $H_{kd+1}$ cube is $kd+1$. So, a lower bound of the congestion is

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

### 3.1.3 Active-degree and node-load

Some nodes at level $k$ of the hyper-pyramid $\tilde{P}(k,d)$ use all edges of the node they are mapped to. For example, consider nodes $a(1,0)$ or $a(1,2^{d-1})$ in Figures 7 and 8. For these nodes all the $d+1$ cube edges are used after the first induction step. For each induction step $d$ new cube edges are used. Hence, $\alpha = kd+1$.

For the node-load $\beta$, consider the following properties:

1. For $k=1$, $\beta_i$ for the spare root is 0, for the real root it is $2^d$, and the maximum $\beta_i$ for $i$ being a node at level one is $\leq \frac{2^{d+2}}{d+1}+d$.
2. A spare root after an induction step is a spare root also before the induction step. The $\beta_i$ for the spare root remains 0.

3. A real root after an induction step is a spare root before the induction step. The $\beta_i$ for the node increases by $2^d$.

4. A node at level one after an induction step is a real root before the induction step. The $\beta_i$ for the node increases by at most $\frac{2^{d+2}}{d+1} + d$.

5. A node at level $i$, $i > 1$, after an induction step is a node at level $i-1$ before the induction step. The $\beta_i$ for the node increases by $d$.

By these properties, one can show by induction that $\beta_i$ of a node at level $k-1$ is maximum, for $k \geq 2$. The node-load

$$\beta \leq \begin{cases} \max\{2^d, \frac{2^{d+2}}{d+1} + d\}, & k = 1, \\ (1 + \frac{4}{d+1})2^d + (k-1)d\}, & k \geq 2. \end{cases}$$

## 3.2 Embedding a $\hat{P}(k,2)$ hyper-pyramid in a Boolean cube with minimal expansion, dilation 2, and congestion 2

### 3.2.1 Dilation and congestion

Clearly, the lower bound of the dilation is 2 for $k \geq 1$ since there exist cycles of odd length in the pyramid. The lower bound of the congestion has been shown to be at least 2 by Lai and White [23] for embedding a $P(k,2,2)$ pyramid in an $H_{2k+1}$ cube for $k \geq 1$. Since a $P(k,2,2)$ pyramid is a subgraph of a $\hat{P}(k,2)$ hyper-pyramid (with the same number of nodes), the lower bound congestion also applies to the hyper-pyramid.

**Theorem 2** *A $\hat{P}(k,2)$ hyper-pyramid can be mapped into an $H_{2k+1}$ Boolean cube with dilation 2 and congestion 2.*

**Proof:** We use the embedding function $f$ defined in the previous section, and define a path of length 2 in the cube for each hyper-pyramid edge of dilation 2.

$$\begin{array}{ll}
a(0,\varepsilon) \rightarrow a(1,0): & a^0(0,\varepsilon) \rightarrow a^0(0',\varepsilon) \\
a(0,\varepsilon) \rightarrow a(1,1): & a^0(0,\varepsilon) \rightarrow a^1(0,\varepsilon) \\
a(0,\varepsilon) \rightarrow a(1,2): & a^0(0,\varepsilon) \rightarrow a^0(0',\varepsilon) \rightarrow a^2(0',\varepsilon) \\
a(0,\varepsilon) \rightarrow a(1,3): & a^0(0,\varepsilon) \rightarrow a^1(0,\varepsilon) \rightarrow a^3(0,\varepsilon) \\
a(1,0) \rightarrow a(1,1): & a^0(0',\varepsilon) \rightarrow a^1(0',\varepsilon) \rightarrow a^1(0,\varepsilon) \\
a(1,0) \rightarrow a(1,2): & a^0(0',\varepsilon) \rightarrow a^2(0',\varepsilon) \\
a(1,1) \rightarrow a(1,3): & a^1(0,\varepsilon) \rightarrow a^3(0,\varepsilon) \\
a(1,2) \rightarrow a(1,3): & a^2(0',\varepsilon) \rightarrow a^3(0',\varepsilon) \rightarrow a^3(0,\varepsilon)
\end{array}$$

The proof is based on induction on $k$ and the induction hypotheses are

1. congestion $\leq 2$ with either one of the two roots of the two-rooted hyper-pyramid selected as the root of the hyper-pyramid,

2. the two roots are mapped to adjacent cube nodes, and

3. the cube edge between the two roots are not used.

Note that in hypothesis 1, congestion is considered separately for the two roots. In considering one root the congestion contributed by the other root is ignored.

**Basis:** For a two-rooted hyper-pyramid $\tilde{P}(0,2)$, all three conditions are satisfied.

**Induction:** Assume that the embedding function $f_n$ extended with assignments of intermediate nodes for paths of length two satisfies the induction hypotheses. $f_{n+1}$ is defined in terms of $f_n$. The two-rooted hyper-pyramid $\tilde{P}(n+1,2)$ embedded in an $H_{2n+3}$ cube is composed of 4 copies of two-rooted hyper-pyramids $\tilde{P}(n,2)$ each embedded in an $H_{2n+1}$ cube. One of the two roots (i.e., the real root) in each copy becomes a node at level 1. The other (i.e., the spare root) either becomes one of the two new roots, or is unused. Next, consider the edges of the newly formed two-rooted hyper-pyramid $\tilde{P}(n+1,2)$ composed of the following three sets:

1. Set $S_1$: the four edges between the new root and its four children; the four edges between the children of the new root.

2. Set $S_2$: the edges between the four subpyramids, except the four edges included in the set $S_1$.

3. Set $S_3$: the edges within the four subpyramids.

Let $\mathcal{F}(S_1)$, $\mathcal{F}(S_2)$ and $\mathcal{F}(S_3)$ be the set of cube edges to which the edges in the sets $S_1$, $S_2$ and $S_3$ are mapped. If we can show that

1. the sets $\mathcal{F}(S_1)$, $\mathcal{F}(S_2)$ and $\mathcal{F}(S_3)$ are disjoint,

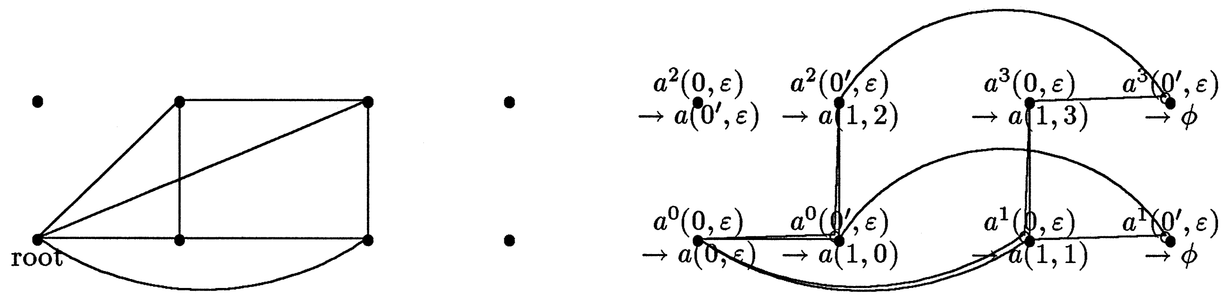
2. each of the sets $\mathcal{F}(S_1)$, $\mathcal{F}(S_2)$ and $\mathcal{F}(S_3)$ gives rise to a congestion $\leq 2$,

3. conditions 2 and 3 of the induction hypotheses are satisfied,

then the proof is complete. Figure 10-(b) shows the embedding of the set $S_1$. The empty node "o" next to a solid node "•" represents an intermediate node for a length-two path. Only the edges in $\mathcal{F}(S_1)$ are shown. For comparison, the logical edges which describe the connection of the hyper-pyramid are shown on the left, Figure 10-(a).

Note that $\mathcal{F}(S_1)$ only contains cube edges of the 3-dimensional cube formed by the eight old roots prior to induction. The set $\mathcal{F}(S_2)$ only contains edges in the last two cube dimensions, i.e., $2n+1$ and $2n+2$, if we label the cube dimensions from 0. But, none of these edges fall in the subcube formed by the old roots, since they are direct connections between nodes at levels greater than 0 in the two-rooted hyper-pyramids $\tilde{P}(n,2)$. From induction hypothesis 3 and the definition of the set $S_3$ it follows that the set $\mathcal{F}(S_3)$ does not contain any edges in that cube either. Hence, $\mathcal{F}(S_1)$ is disjoint with respect to the sets $\mathcal{F}(S_2)$ and $\mathcal{F}(S_3)$. Since $\mathcal{F}(S_2)$ only contains edges in the last two dimensions, but $\mathcal{F}(S_3)$ does not contain any edges in these dimensions $\mathcal{F}(S_2)$ and $\mathcal{F}(S_3)$ are also disjoint.

From the definition of the embedding function $f$ and the path selection indicated in Figure 10, it follows that $cong(\mathcal{F}(S_1)) \leq 2$. The definition of the embedding function immediately gives




(a) logical edges of hyper-pyramid

(b) physical edges/paths in cube

Figure 10: (a) The set of edges $S_1$ of hyper-pyramids. (b) The set of edges $\mathcal{F}(S_1)$ of cube.

$cong(\mathcal{F}(S_2)) = 1$. From condition 1 of the induction hypotheses it follows that $cong(\mathcal{F}(S_3))$ is preserved. Hence, congestion $\leq 2$ is preserved since the sets $\mathcal{F}(S_1)$, $\mathcal{F}(S_2)$ and $\mathcal{F}(S_3)$ are disjoint. Condition 2 of the induction hypothesis follows directly from the definition of the embedding function $f_{n+1}$, see Figure 10. Condition 3 follows from the definition of the embedding function $f$ and the path selection as given in Figure 10. None of the sets $\mathcal{F}(S_1)$, $\mathcal{F}(S_2)$ and $\mathcal{F}(S_3)$ contain the cube edge between the two new roots. ∎

### 3.2.2 Active-degree and node-load

The embedding function described here is a special case of the modified function described in Section 3.1. Since $\alpha = kd + 1$ in Section 3.1, we have $\alpha = 2k + 1$ here. By substituting $d = 2$ in the node-load derived in Section 3.1, we have

$$\beta \leq \begin{cases} 8, & k = 1, \\ 2k + 8, & k \geq 2. \end{cases}$$

However, a slightly tighter bound for $f$ with the path specification given in this section can be proved to be

$$\beta = \begin{cases} 5, & k = 1, \\ 2k + 5, & k \geq 2, \end{cases}$$

by induction using the follow facts:

1. For $k = 1$, the maximum $\beta_i$ for a node $i$ being the spare root, the real root and a node at level one are 0, 4 and 5, respectively.

2. A spare root after an induction step is a spare root before the induction step. The $\beta_i$ for the spare root remains 0.

3. A real root after an induction step is a spare root before the induction step. The $\beta_i$ for the node increases by 4 after the induction step.

4. A node at level one after an induction step is the real root before the induction step. The $\beta_i$ for the node increases by 5 after the induction step for the path selection made in this section (compared to an increase of 8 for the embedding for arbitrary $d$).

5. A node at level $i$, $i > 1$, after an induction step is a node at level $i-1$ before the induction step. The $\beta_i$ for the node increases by 2 after the induction step.

### 3.3 Embedding one hyper-pyramid $\hat{P}(k,d)$ and $(2^d-2)$ hyper-pyramids $\hat{P}(k-1,d)$'s in a Boolean cube with expansion $\approx 1$ and dilation $d+1$

Even though minimal expansion (i.e., expansion $< 2$) is achieved in Section 3.1, $2^d - 2$ cube nodes are not used in each induction step. It is possible, however, to embed a $\hat{P}(k,d)$ hyper-pyramid and $2^d - 2$ smaller hyper-pyramids $\hat{P}(k-1,d)$ into an $H_{kd+1}$ cube at the same time, such that only one cube node is not used.

#### 3.3.1 Dilation

**Theorem 3** *A hyper-pyramid $\hat{P}(k,d)$ together with $(2^d-2)$ hyper-pyramids $\hat{P}(k-1,d)$, $k \geq 1$ and $d \geq 2$, can be embedded in an $H_{kd+1}$ Boolean cube with expansion $\approx 1$ (only one cube node is not used) and dilation $d+1$.*

**Proof:** In the following, the subscript on $\tilde{P}$, $\hat{P}$, and $a$ is used to identify different hyper-pyramids and vertices therein. For notational convenience, we let $a_1(0,\varepsilon)$ denote $a_0(0',\varepsilon)$. For the proof we consider one two-rooted hyper-pyramid $\tilde{P}(k,d)$ and $2^d-2$ hyper-pyramids $\hat{P}(k-1,d)$ (with single roots). Let the mapping function be $f_k$. The proof is by induction and the hypotheses are that for $k \leq n$ the following two conditions hold:

1. A two-rooted hyper-pyramid $\tilde{P}_0(k,d)$ and $(2^d-2)$ hyper-pyramids $\hat{P}_j(k-1,d)$, $2 \leq j < 2^d$, $k \geq 1$ and $d \geq 2$, can be embedded in an $H_{kd+1}$ Boolean cube with dilation $d+1$.
2. *Hamming*$(f_k(a_x(0,\varepsilon)), f_k(a_{x\{m\}}(0,\varepsilon))) = 1$ for all $0 \leq x < 2^d$, $0 \leq m < d$, i.e., all the $2^d$ roots are mapped to an $H_d$ subcube in the $H_{kd+1}$ cube, and the two roots of $\tilde{P}_0$ are mapped to adjacent cube nodes.

**Basis:** For $k=1$, the two-rooted hyper-pyramid $\tilde{P}_0(1,d)$ contains the roots $a_0(0,\varepsilon)$ and $a_1(0,\varepsilon)$ and the base $H_d$ ($a_0(1,j), 0 \leq j < 2^d$). For each of the $(2^d-2)$ hyper-pyramids $\hat{P}_x(0,d)$, $x \in \{2,3,\ldots,2^d-1\}$, $\hat{P}_x(0,d)$ is the root node. Define $f_1$ as:

$$f_1(a_j(0,\varepsilon)) = j||0, \qquad \forall 0 \leq j < 2^d,$$

$$f_1(a_0(1,j)) = j||1, \qquad \forall 0 \leq j < 2^d.$$

It is easily seen that $f_1$ satisfies the two conditions. Figures 11 and 12 show the mapping for $d=2$ and $d=3$, respectively, and $k=1$.

**Induction:** Assume that the mapping $f_n$ satisfies the above two conditions. Consider a $H_{(n+1)d+1}$ cube with hyper-pyramids embedded by the embedding function $f_n$ in each of the $H_d$ cubes of dimension $nd+1$. We define $f_{n+1}$ in terms of $f_n$ by the following rules:

$$f_{n+1}(a_\ell(0,\varepsilon)) \;=\; f_n(a_0^\ell(0,\varepsilon)),$$

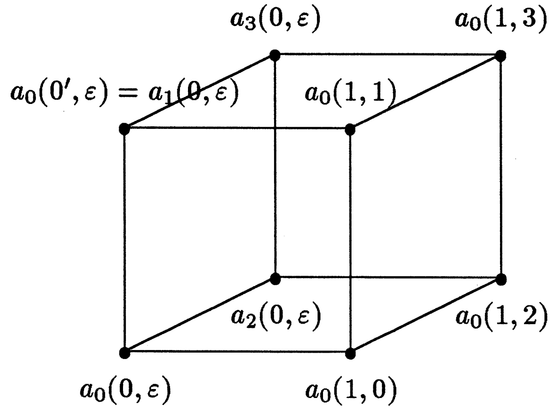



Figure 11: The basis: A two-rooted hyper-pyramid $\tilde{P}_0(1,2)$, a hyper-pyramid $\hat{P}_2(0,2)$ and a hyper-pyramid $\hat{P}_3(0,2)$ embedded in an $H_3$ cube with dilation 3.

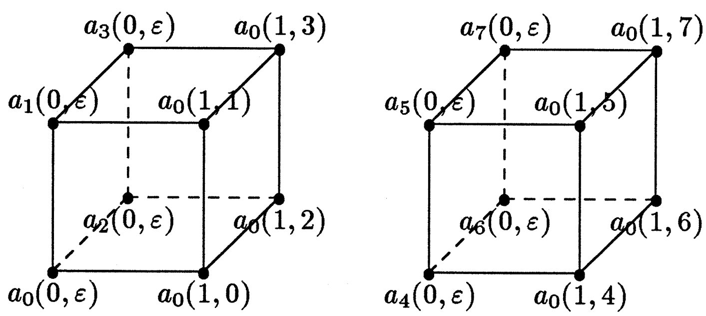



Figure 12: The basis: A two-rooted hyper-pyramid $\tilde{P}_0(1,3)$ and 6 hyper-pyramids $\hat{P}_j(0,3)$'s, $j \in \{2,3,\ldots,7\}$, in an $H_4$ cube with dilation 4.

$$
\begin{aligned}
f_{n+1}(a_0(1,\ell)) &= f_n(a_1^\ell(0,\varepsilon)), \\
f_{n+1}(a_0(i,\ell||j)) &= f_n(a_0^\ell(i-1,j)), \qquad 2 \le i \le n+1, \\
f_{n+1}(a_x(i,\ell||j)) &= f_n(a_{x\oplus\gamma(\ell,x)}^\ell(i-1,j)), \qquad 1 \le i \le n+1,\ 2 \le x < 2^d,
\end{aligned}
$$

where $\gamma(\ell, x) = (\gamma_{d-1}\gamma_{d-2}\ldots\gamma_0)$. The value of $\gamma$ is determined from $\ell$ and $x$ as follows: if $\ell_m = 0$ and $x_{d-1}x_{d-2}\ldots x_{m+1} \neq 0$, then $\gamma_m = 1$, else $\gamma_m = 0$, $0 \le m < d-1$. The superscript $\ell$ identifies nodes of different hyper-pyramids mapped to distinct $H_{nd+1}$ cubes, as before. So, $f_n(a_x^\ell(i,j)) = \ell||f_n(a_x(i,j))$. By rule 1 of the recursive definition above, we select the root $a_0(0,\varepsilon)$ of $\tilde{P}_0(n,d)$ to be the spare root. In the induction $2^d$ cubes with embedded hyper-pyramids are used to form a new embedding. The number of spare roots in the $2^d$ cubes are $2^d$. Two of them serve as the two new roots of $\tilde{P}_0(n+1,d)$ and the remaining $2^d-2$ spare roots serve as the new roots (one for each) of the $2^d-2$ $\hat{P}(n,d)$ hyper-pyramids. (For notational convenience, we choose the two new roots of $\tilde{P}(n+1,d)$ from cubes 0 and 1, instead of choosing from cubes 0 and $2^{d-1}$ as in Section 3.1.) By rule 2, we select $a_1(0,\varepsilon)$ as the real root of the two-rooted hyper-pyramid $\tilde{P}_0(n,d)$ in each $H_{nd+1}$ cube, i.e., it becomes a node at level 1 of $\tilde{P}_0(n+1,d)$ hyper-pyramid. Rule 3 moves nodes of $\tilde{P}_0(n,d)$ at level $i-1$ to nodes of $\tilde{P}_0(n+1,d)$ at levels $i \ge 2$. Rule 4 moves nodes of the $\hat{P}_x(n-1,d)$ hyper-pyramids, $2 \le x < 2^d$, at levels $i-1$ to nodes of the $\hat{P}_x(n,d)$ hyper-pyramids at level $i$. Note that rule 4 is complicated by the exchange between adjacent hyper-pyramids as defined by $\gamma$. For example, for $d=3$ and $\ell=0$, $\gamma = 001, 001, 011, 011, 011$ and $011$ for $x = 2,3,4,5,6$ and $7$, respectively. A naive embedding without exchange, i.e., $\gamma = 0$, would have dilation $2d$ for some hyper-pyramid.

Before proving that $f_{n+1}$ is well-defined and also satisfies the induction hypotheses, we give some examples of the induction step. Figure 13 shows the induction step for $d=2$. For clarity, only the 4 interesting root nodes are shown in each $H_{nd+1}$ cube, and the corresponding connections between the four cubes are omitted. The two arrowed lines denote that the roles of $\hat{P}_2$ and $\hat{P}_3$ are exchanged in even cubes. Figure 14 shows the edge dilation of the newly formed edges of the first two levels. For convenience, we draw two instances of each of the nodes $a_0(1,j)$, $0 \le j < 4$. Multiple instances refer to the same cube node. The solid, dashed and dotted lines in the figure represent Hamming distances of 1, 2 and 3, respectively. Note that by rule 4, $\hat{P}_2(n,2)$ and $\hat{P}_3(n,2)$ embedded in cubes 0 and 2 are exchanged (within the same cube). Without these exchanges, the dilation for $\hat{P}_3$ would have been 4.

Figure 15 shows the induction step of the naive embedding ($\gamma = 0$) for $d=3$. The dilation ranges from $d+1$ to $2d$ depending on hyper-pyramid. With the exchanges defined by $\gamma$, the embedding is shown in Figure 16. The exchange is indicated by a two-way arrow. Tables 1 and 2 give the dilations of various edges and dilations of different hyper-pyramids $\hat{P}_j$ for the embeddings shown in Figures 15 and 16, respectively.

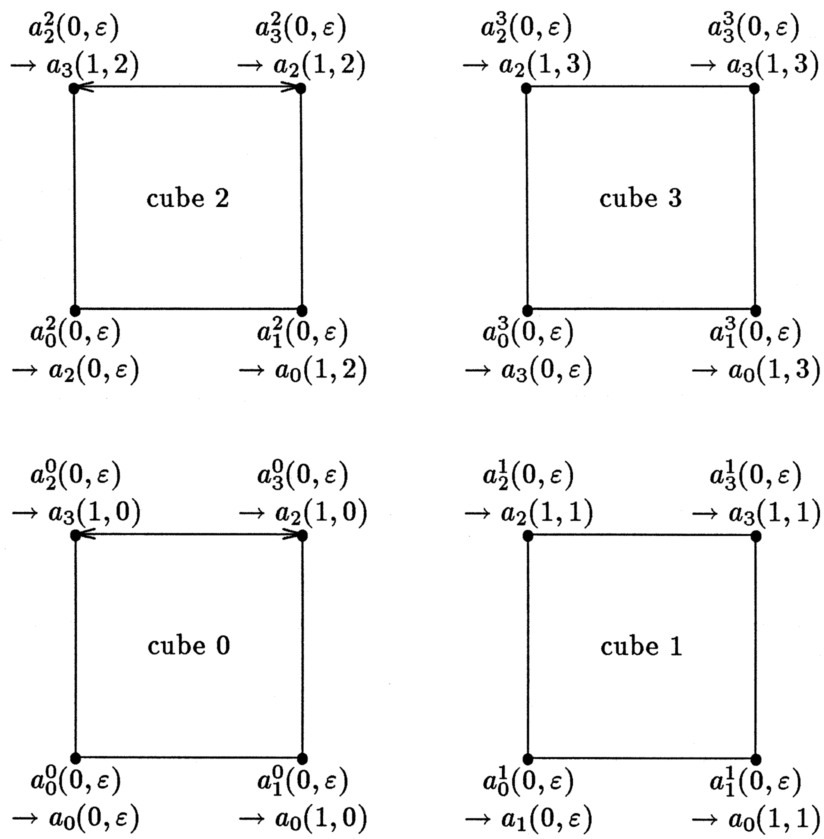



Figure 13: The induction: Forming a two-rooted $\tilde{P}_0(n+1,2)$, a $\hat{P}_2(n,2)$ and a $\hat{P}_3(n,2)$ out of four copies of "a two-rooted $\tilde{P}_0(n,2)$, a $\hat{P}_2(n-1,2)$ and a $\hat{P}_3(n-1,2)$". The two arrowed lines denote that the roles of $\hat{P}_2$ and $\hat{P}_3$ are exchanged in even cubes.

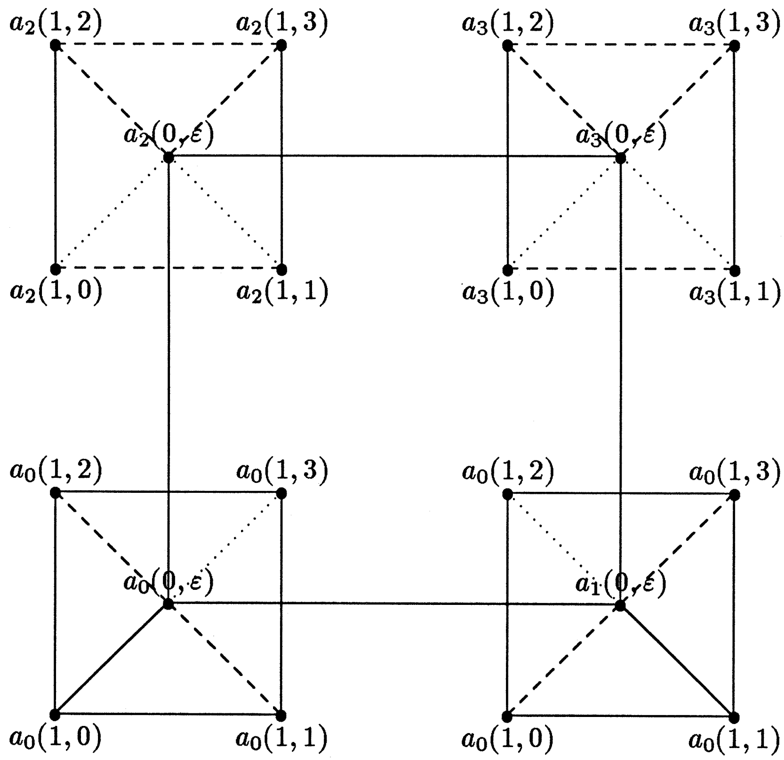



Figure 14: The edge dilation of the newly formed edges of nodes in the first two levels. The solid, dashed and dotted lines in the figure represent Hamming distances (edge dilations) of 1, 2 and 3, respectively, in cube.

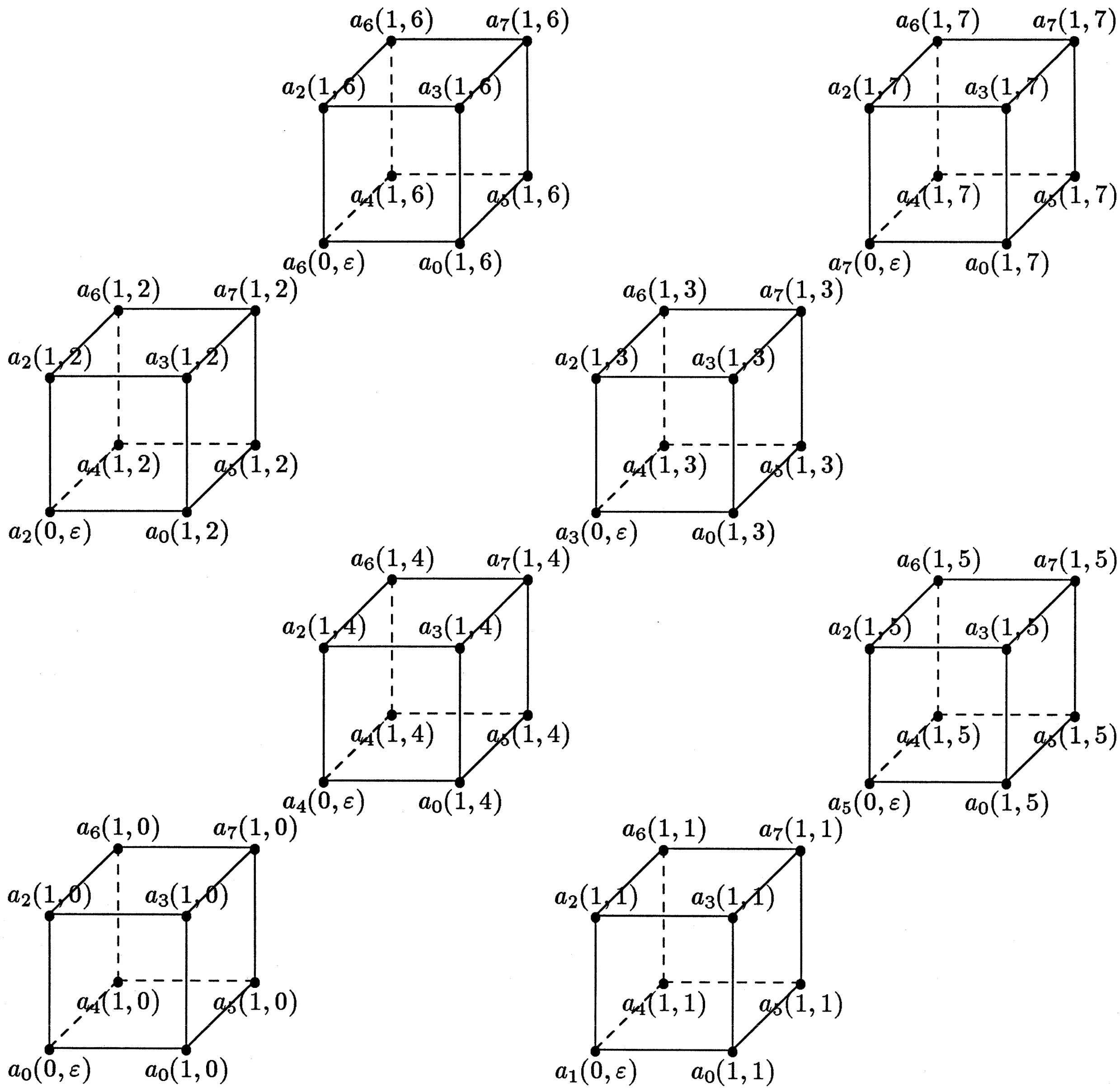



Figure 15: The induction step of the naive embedding ($\gamma = 0$) for $d = 3$. The dilation is 6.

| $a_j(0,\varepsilon)$ | $a_j(1,0)$ | $a_j(1,1)$ | $a_j(1,2)$ | $a_j(1,3)$ | $a_j(1,4)$ | $a_j(1,5)$ | $a_j(1,6)$ | $a_j(1,7)$ | dilation |
|---|---|---|---|---|---|---|---|---|---|
| $a_0(0,\varepsilon)$ | 1 | 2 | 2 | 3 | 2 | 3 | 3 | 4 | 4 |
| $a_1(0,\varepsilon)$ | 2 | 1 | 3 | 2 | 3 | 2 | 4 | 3 | 4 |
| $a_2(0,\varepsilon)$ | 2 | 3 | 1 | 2 | 3 | 4 | 2 | 3 | 4 |
| $a_3(0,\varepsilon)$ | 4 | 3 | 3 | 2 | 5 | 4 | 4 | 3 | 5 |
| $a_4(0,\varepsilon)$ | 2 | 3 | 3 | 4 | 1 | 2 | 2 | 3 | 4 |
| $a_5(0,\varepsilon)$ | 4 | 3 | 5 | 4 | 3 | 2 | 4 | 3 | 5 |
| $a_6(0,\varepsilon)$ | 4 | 5 | 3 | 4 | 3 | 4 | 2 | 3 | 5 |
| $a_7(0,\varepsilon)$ | 6 | 5 | 5 | 4 | 5 | 4 | 4 | 3 | 6 |

Table 1: The Hamming distance between the root and its children for the naive embedding.

| $a_j(0,\varepsilon)$ | $a_j(1,0)$ | $a_j(1,1)$ | $a_j(1,2)$ | $a_j(1,3)$ | $a_j(1,4)$ | $a_j(1,5)$ | $a_j(1,6)$ | $a_j(1,7)$ | dilation |
|---|---|---|---|---|---|---|---|---|---|
| $a_0(0,\varepsilon)$ | 1 | 2 | 2 | 3 | 2 | 3 | 3 | 4 | 4 |
| $a_1(0,\varepsilon)$ | 2 | 1 | 3 | 2 | 3 | 2 | 4 | 3 | 4 |
| $a_2(0,\varepsilon)$ | 3 | 3 | 2 | 2 | 4 | 4 | 3 | 3 | 4 |
| $a_3(0,\varepsilon)$ | 3 | 3 | 2 | 2 | 4 | 4 | 3 | 3 | 4 |
| $a_4(0,\varepsilon)$ | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 3 | 4 |
| $a_5(0,\varepsilon)$ | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 3 | 4 |
| $a_6(0,\varepsilon)$ | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 3 | 4 |
| $a_7(0,\varepsilon)$ | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 3 | 4 |

Table 2: The Hamming distance between the root and its children for the improved embedding.

We now first prove that the recursive definition is "well-defined". By "well-defined", we mean that "if $f_{n+1}(a_x(i,\ell||j)) = f_{n+1}(a_{x'}(i',\ell'||j'))$, then $x = x'$, $i = i'$, $\ell = \ell'$ and $j = j'$. This is obvious if $\gamma(\ell, x) \equiv 0$. With $\gamma$ a function of $\ell$ and $x$, it suffices to prove that $f_{n+1}(a_{x\oplus\gamma(\ell,x)}(i,\ell||j)) = f_n(a_x^\ell(i-1,j))$. From the definition $f_{n+1}(a_{x\oplus\gamma(\ell,x)}(i,\ell||j)) = f_n(a^\ell_{x\oplus\gamma(\ell,x)\oplus\gamma(\ell,x\oplus\gamma(\ell,x))}(i-1,j))$. So, we simply prove that $\gamma(\ell,x) = \gamma(\ell, x\oplus\gamma(\ell,x))$. This is true by Lemma 11 in Appendix A.

We now prove that the recursive definition satisfies the induction hypotheses. Condition 2 of the induction hypotheses is preserved due to rule 1 in the definition of $f_{n+1}$. In order to prove that condition 1 holds for $k = n+1$, we partition the newly formed hyper-pyramid 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 2. Firstly, 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 $f_{n+1}(a_x(i,\ell||j))$ and $f_{n+1}(a_x(i,\ell^{\{m\}}||j))$.

$$\begin{aligned}
\text{The former term} &= f_n(a^\ell_{x\oplus\gamma(\ell,x)}(i-1,j)) \\
&= \ell||f_n(a_{x\oplus\gamma(\ell,x)}(i-1,j)), \text{ and} \\
\text{the latter term} &= f_n(a^{\ell^{\{m\}}}_{x\oplus\gamma(\ell^{\{m\}},x)}(i-1,j))
\end{aligned}$$

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

Let $\gamma(\ell, x) = y$. Then, from the definition of $\gamma(\ell, x)$, one can derive $\gamma(\ell^{\{m\}}, x) = y$ or $y^{\{m\}}$. So, the dilation in $S_2$ is either one or two. The dilation two occurs when there is an exchange operation involved in one side of the cubes. To determine the edge dilation in $S_1$, we consider the subsets: $S_{11}$, the edges between nodes at level 1, and $S_{12}$, the edges between roots and nodes at level 1. The edge dilation in $S_{11}$ is either 1 or 2 for the same reasons as the dilation of edges in the set $S_2$ is at most 2. For the edge dilation in $S_{12}$, consider $Hamming(f_{n+1}(a_x(0,\varepsilon)), f_{n+1}(a_x(1,\ell)))$. It is $||\ell|| + 1 \leq d+1$, if $x = 0$ or 1. For $x \neq 0$ and $x \neq 1$, $f_{n+1}(a_x(0,\varepsilon)) = f_n(a_0^x(0,\varepsilon))$ and $f_{n+1}(a_x(1,\ell)) = f_n(a^{\ell}_{x\oplus\gamma(\ell,x)}(0,\varepsilon))$. So, the Hamming distance is $||x \oplus \gamma(\ell,x)|| + ||x \oplus \ell||$, which is at most $d+1$ by Lemma 12 in Appendix A. ∎

### 3.3.2 Congestion

For the congestion we show that the dilation $d+1$ embedding yields a congestion of at most $(d+1)(\frac{2^{d+1}}{d+2}+1)$. However, a dilation $2d$ embedding yields a congestion of $\lceil\frac{2^d}{d}\rceil + \frac{2^{d+1}}{d+2} + 1$, which is the same as the congestion for the embedding of a single hyper-pyramid in the same-sized Boolean cube. The following two lemmas describe path assignments and prove the corresponding congestion. Intuitively, in Lemma 9 the paths are selected by first routing between the $2^d$ cubes of an induction step and then routing within these cubes in going from a root to its children, Figure 16. In Lemma 10 routes are selected by first routing within the cubes, then between them.

**Lemma 9** *The congestion for the dilation $d+1$ embedding in Theorem 3 is $(d+1)(\frac{2^{d+1}}{d+2}+1)$.*

**Proof:** We give a sketch of an inductive proof. Let the path from $a_i(0,\varepsilon)$ to $a_i(1,j)$ be passing through an intermediate node $a_j(0,\varepsilon)$. For example in Figure 16, the path from $a_3(0,\varepsilon)$ to $a_3(1,4)$ goes through the intermediate node $a_4(0,\varepsilon)$. Note that there are $2^d$ roots and each root has paths to $2^d$ intermediate nodes. The $2^d$ roots are the same set of nodes as the $2^d$ intermediate nodes. Consider the first half of all the paths from the $2^d$ roots of the form $a_i(0,\varepsilon)$ to all the $2^d$ intermediate nodes. This is the same as embedding $2^d$ flat trees of size $2^d$ rooted at different nodes in an $H_d$ cube. By Lemma 5, it is bounded from above by $\frac{d2^{d+1}}{d+2} + d$. For the second half of all the paths, it is the same as embedding a single flat tree in an $H_d$ cube, which is bounded from above by $\frac{2^{d+1}}{d+2}$ by Lemma 3. The inductive hypotheses are that the edge congestion of the new $d$ cube dimensions is at most $\frac{d2^{d+1}}{d+2} + d$ (i.e., the first half of the paths). During an induction step, edges of the considered $d$ cube dimensions will increase their edge congestions by at most $\frac{2^{d+1}}{d+2} + 1$ (i.e., the second half of the paths plus the edge introduced by the exchange operation $\gamma$). Note that the paths assignment for the basis, see for example Figure 12, can be done such that the edge congestion is 1 for edges of dimensions 0, and is at most $\frac{2^{d+1}}{d+2}$ for edges of dimensions 1 to $d$. ∎

**Lemma 10** *The congestion for the dilation $2d$ embedding in Theorem 3 is $\lceil\frac{2^d}{d}\rceil + \frac{2^{d+1}}{d+2} + 1$.*

**Proof:** We give a sketch of an inductive proof. We choose the naive embedding which has dilation $2d$, i.e., $\gamma = 0$ as in Figure 15. In assigning the paths, some of them are not shortest paths, but do not affect the dilation. Let the path from $a_i(0, \varepsilon)$ to $a_i(1, j)$ go through the intermediate node $a_i(1, i)$. For the second half of all the paths, the congestion is $\leq \frac{2^{d+1}}{d+2}$ due to Lemma 3. For the first half of the paths, $2^d$ paths need to be embedded between a pair of nodes in an $H_d$ cube. At the expense of increasing the dilation by two when the distance between the pair of nodes is less than $d$, the congestion is $\lceil \frac{2^d}{d} \rceil$. This congestion is realized by making use of the $d$ edge-disjoint paths between any two nodes in an $H_d$ cube [30]. The length of the first-half paths is $\leq d + 1$, and the length of the second-half paths is $\leq d$. However, only one of the second-half paths is of length $d$, and not all of the first-half paths are of length $d+1$. The paths can be paired up such that the length is $\leq 2d$. The inductive hypotheses are that the edge congestion of the newest $d$ cube dimensions is at most $\frac{2^{d+1}}{d+2} + 1$ (i.e., the second half of the paths plus the edge between nodes of level one). During an induction step, edges of the considered $d$ cube dimensions will increase their edge congestions by at most $\lceil \frac{2^d}{d} \rceil$ (i.e., the first half of the paths). ∎

### 3.3.3 Active-degree and node-load

Some nodes at level $k$ of the hyper-pyramid $\hat{P}_0(k, d)$ use all edges of the cube nodes to which they are mapped. For example, consider node $a_0(1, 0)$ or $a_0(1, 1)$ in Figures 11 and 12. Each uses all the $d + 1$ cube edges after the first induction step, and each uses all the $d$ new cube edges after every additional induction step. Hence, $\alpha = kd + 1$.

For the node-load, we follow the path assignments in the proof of Lemma 10, which is of dilation $2d$ and congestion $\lceil \frac{2^d}{d} \rceil + \frac{2^{d+1}}{d+2} + 1$. For convenience, we use "large root", "small root", "large node" and "small node" to represent "the real root of the large hyper-pyramid", "the root of a small hyper-pyramid", "a node of the large hyper-pyramid" and "a node of a small hyper-pyramid", respectively. We show that the node-load

$$\beta = \begin{cases} 2^d, & k = 1, \\ 3 \cdot 2^d, & k = 2, \\ (3 + \frac{4}{d+2})2^d + (k-3)d, & k \geq 3, \end{cases}$$

by induction based on the following properties:

1. For $k = 1$, the maximum $\beta_i$ for a node $i$ being the spare root, the large root, one of the small roots, and one of the large nodes at level one are $\frac{2^{d+2}}{d+2}$, $2^d$, $\frac{2^{d+2}}{d+2}$ and $d+1$, respectively.
2. A spare root after an induction step is a spare root before the induction step. The node-load for the node does not increase during an induction step.
3. A (large or small) root after an induction step is a spare root before the induction step. The node-load for the node increases by $2^d$ during an induction step.
4. A large (small) node at level one after an induction step is a large (small) root before the induction step. The node-load for the node increases at most by $2^{d+1}$.

5. A large (small) node at level $i$, $i > 1$, after an induction step is a large (small) node at level $i-1$ before the induction step. The node-load for the node increases by $d$.

The node-load $\beta_i$ is dominated by a node at level $k-2$ for $k \geq 3$.

For the dilation $d+1$ and congestion $O(2^d)$ embedding, we follow the path assignments described in the proof of Lemma 9 except that Lemma 6 is applied in the proof instead, i.e., no path is passing through the spare root. The congestion remains of the same order. The node-load can be bounded from above by $O(d2^d) + O(kd)$. The proof is based on the following properties:

1. For $k = 1$, the maximum node-load $\beta_i$ for a node $i$ being the spare root, the large root, one of the small roots, and one of the large nodes at level one are $\frac{2^{d+2}}{d+2}$, $2^d$, $\frac{2^{d+2}}{d+2}$ and $d+1$, respectively.
2. A spare root after an induction step is a spare root before the induction step. The node-load for the node does not increase during the induction step.
3. A (large or small) root after an induction step is a spare root before the induction step. The $\beta_i$ for the node increases by $O(d2^d)$ during the induction step. This is because the edge congestion of the $d$ edges in the $d$ new dimensions is of order $O(2^d)$, Lemma 5.
4. A large (small) node at level one after an induction step is a large (small) root before the induction step. The $\beta_i$ for the node increases at most by $O(\frac{2^d}{d})$.
5. A large (small) node at level $i$, $i > 1$, after an induction step is a large (small) node at level $i-1$ before the induction step. The $\beta_i$ for the node increases by $O(d)$.

## 3.4 Embedding one hyper-pyramid $\hat{P}(k,2)$ and two hyper-pyramids $\hat{P}(k-1,2)$'s in a Boolean cube with expansion $\approx 1$, dilation 3 and congestion 3

### 3.4.1 Dilation and congestion

In this section, we show that for $d = 2$ the results in the previous section can be specialized so that the congestion is at most 3.

**Theorem 4** *A hyper-pyramid $\hat{P}(k,2)$ and two hyper-pyramids $\hat{P}(k-1,2)$, $k \geq 1$, can be embedded in an $H_{2k+1}$ Boolean cube with expansion $\approx 1$, dilation 3 and congestion 3. (Note that congestion 3 is obtained by considering all three hyper-pyramids simultaneously.)*

**Proof:** We use the same embedding function $f$ as for the $\hat{P}(k,d)$ hyper-pyramids in Theorem 3. In addition, we define all the intermediate nodes of paths generated by hyper-pyramid edges.

$$\begin{aligned}
a_0(0,\varepsilon) \to a_1(1,0):&\quad a_0^0(0,\varepsilon) \to a_1^0(0,\varepsilon)\\
a_0(0,\varepsilon) \to a_1(1,1):&\quad a_0^0(0,\varepsilon) \to a_1^0(0,\varepsilon) \to a_1^1(0,\varepsilon)\\
a_0(0,\varepsilon) \to a_1(1,2):&\quad a_0^0(0,\varepsilon) \to a_1^0(0,\varepsilon) \to a_1^2(0,\varepsilon)
\end{aligned}$$

$$
\begin{array}{ll}
a_0(0,\varepsilon) \to a_1(1,3): & a_0^0(0,\varepsilon) \to a_0^2(0,\varepsilon) \to a_0^3(0,\varepsilon) \to a_1^3(0,\varepsilon) \\
a_0(1,0) \to a_0(1,1): & a_1^0(0,\varepsilon) \to a_1^1(0,\varepsilon) \\
a_0(1,1) \to a_0(1,3): & a_1^1(0,\varepsilon) \to a_1^3(0,\varepsilon) \\
a_0(1,3) \to a_0(1,2): & a_1^3(0,\varepsilon) \to a_1^2(0,\varepsilon) \\
a_0(1,2) \to a_0(1,0): & a_1^2(0,\varepsilon) \to a_1^0(0,\varepsilon)
\end{array}
$$

$$
\begin{array}{ll}
a_2(0,\varepsilon) \to a_2(1,0): & a_0^2(0,\varepsilon) \to a_1^2(0,\varepsilon) \to a_3^2(0,\varepsilon) \to a_3^0(0,\varepsilon) \\
a_2(0,\varepsilon) \to a_2(1,1): & a_0^2(0,\varepsilon) \to a_2^2(0,\varepsilon) \to a_2^0(0,\varepsilon) \to a_2^1(0,\varepsilon) \\
a_2(0,\varepsilon) \to a_2(1,2): & a_0^2(0,\varepsilon) \to a_1^2(0,\varepsilon) \to a_3^2(0,\varepsilon) \\
a_2(0,\varepsilon) \to a_2(1,3): & a_0^2(0,\varepsilon) \to a_2^2(0,\varepsilon) \to a_2^3(0,\varepsilon) \\
a_2(1,0) \to a_2(1,1): & a_3^0(0,\varepsilon) \to a_3^1(0,\varepsilon) \to a_2^1(0,\varepsilon) \\
a_2(1,1) \to a_2(1,3): & a_2^1(0,\varepsilon) \to a_2^3(0,\varepsilon) \\
a_2(1,3) \to a_2(1,2): & a_2^3(0,\varepsilon) \to a_3^3(0,\varepsilon) \to a_3^2(0,\varepsilon) \\
a_2(1,2) \to a_2(1,0): & a_3^2(0,\varepsilon) \to a_3^0(0,\varepsilon)
\end{array}
$$

$$
\begin{array}{ll}
a_3(0,\varepsilon) \to a_3(1,0): & a_0^3(0,\varepsilon) \to a_2^3(0,\varepsilon) \to a_2^1(0,\varepsilon) \to a_2^0(0,\varepsilon) \\
a_3(0,\varepsilon) \to a_3(1,1): & a_0^3(0,\varepsilon) \to a_1^3(0,\varepsilon) \to a_3^3(0,\varepsilon) \to a_3^1(0,\varepsilon) \\
a_3(0,\varepsilon) \to a_3(1,2): & a_0^3(0,\varepsilon) \to a_2^3(0,\varepsilon) \to a_2^2(0,\varepsilon) \\
a_3(0,\varepsilon) \to a_3(1,3): & a_0^3(0,\varepsilon) \to a_1^3(0,\varepsilon) \to a_3^3(0,\varepsilon) \\
a_3(1,0) \to a_3(1,1): & a_2^0(0,\varepsilon) \to a_3^0(0,\varepsilon) \to a_3^1(0,\varepsilon) \\
a_3(1,1) \to a_3(1,3): & a_3^1(0,\varepsilon) \to a_3^3(0,\varepsilon) \\
a_3(1,3) \to a_3(1,2): & a_3^3(0,\varepsilon) \to a_3^2(0,\varepsilon) \to a_2^2(0,\varepsilon) \\
a_3(1,2) \to a_3(1,0): & a_2^2(0,\varepsilon) \to a_2^0(0,\varepsilon)
\end{array}
$$

We prove the theorem by induction on $k$. The induction hypotheses are

1. congestion $\leq 3$,
2. $cong(e) \leq 2$ for all edges $e$ in cube dimension $2k-1$, i.e., the dimension between cube $*0$ and cube $*1$ of the previous induction step, where "$*$" is 0 or 1.
3. $cong((f_k(a_0(0,\varepsilon)), f_k(a_0(0',\varepsilon)))) = 0$,
4. $cong((f_k(a_2(0,\varepsilon)), f_k(a_3(0,\varepsilon)))) \leq 1$,
5. $cong((f_k(a_0(0,\varepsilon)), f_k(a_2(0,\varepsilon)))) \leq 1$, and
6. $cong((f_k(a_0(0',\varepsilon)), f_k(a_3(0,\varepsilon)))) \leq 1$.

Note that in order to determine the congestion, both roots of $\tilde{P}_0$ need to be considered. However, they are considered separately. Figure 17 shows the embedding with the path assignment for the basis $k=1$. The notation of "o" is the same as in Figure 10. It is easily seen that all the 6 conditions of the induction hypothesis are satisfied for $k=1$. Due to the symmetricity of the two roots, $a_0(0,\varepsilon)$ and $a_1(0,\varepsilon)$, the congestion for either one being selected as root for $\hat{P}(k,d)$ is the same. Now, assume the 6 conditions of the induction hypotheses are satisfied for $k=n$. Consider the three disjoint sets of hyper-pyramid edges $S_1$, $S_2$ and $S_3$ of the newly formed hyper-pyramids for $k=n+1$ as defined before, and $\mathcal{F}(S_1)$, $\mathcal{F}(S_2)$ and $\mathcal{F}(S_3)$ the corresponding cube edge sets after the embedding. Figure 18 shows the embedding for the induction step of edges in $\mathcal{F}(S_1)$. For clarity, $\tilde{P}_0$ is shown in (a), $\hat{P}_2$ in (b) and $\hat{P}_3$ in (c). The congestion of each cube edge for the induction step is shown by a label on the edge in Figure 18-(d). (d) is derived by overlapping (a), (b) and (c). Figure 19 shows the assignment of length-two paths in $\mathcal{F}(S_2)$.

We now show that the 6 conditions also hold for $k = n + 1$. From Figure 18-(d), conditions 3 to 6 are easily seen to be satisfied for $k = n + 1$. As for conditions 1 and 2, consider the following facts:

- $\mathcal{F}(S_1)$ contains cube edges of dimensions $2n - 1$, $2n$, $2n + 1$ and $2n + 2$. Further, they are the cube edges of the $H_4$ cube formed by the 16 old roots, Figure 18.
- $\mathcal{F}(S_2)$ contains cube edges of dimensions $2n$, $2n + 1$ and $2n + 2$. Further, if $e = (u, v) \in \mathcal{F}(S_2)$, then the two cube nodes $u$ and $v$ correspond to hyper-pyramid nodes at levels greater than one.
- $\mathcal{F}(S_3)$ contains cube edges of dimensions 0 to $2n$.

Clearly, $\mathcal{F}(S_1) \cap \mathcal{F}(S_2) = \phi$ from the first two facts. Then, condition 2 also holds for $k = n+1$ by the edge congestion of dimension $2n + 1$, see Figures 18-(d) and 19, and the fact that $\mathcal{F}(S_1)$ and $\mathcal{F}(S_2)$ are disjoint. In proving that condition 1 is preserved, consider the following:

- *cong*$(e) \leq 3$ for any $e \in \mathcal{F}(S_1) \cap \mathcal{F}(S_3)$: $\mathcal{F}(S_1) \cap \mathcal{F}(S_3)$ only contains cube edges of dimensions $2n-1$ and $2n$. Consider the congestion of the edges $(f_n(a_0(0, \varepsilon)), f_n(a_0(0', \varepsilon)))$, $(f_n(a_2(0, \varepsilon)), f_n(a_3(0, \varepsilon)))$, $(f_n(a_0(0, \varepsilon)), f_n(a_2(0, \varepsilon)))$, $(f_n(a_0(0', \varepsilon)), f_n(a_3(0, \varepsilon)))$, in each of the 4 cubes from the sets $\mathcal{F}(S_1)$ and $\mathcal{F}(S_3)$. They are at most 3, 1, 2 and 2 for $\mathcal{F}(S_1)$ as seen from Figure 18-(d), and at most 0, 1, 1 and 1 due to conditions 3 to 6 of the induction hypotheses for $k = n$. Therefore, the edge congestion of these 4 cube edges in each cube is at most 3.
- *cong*$(e) \leq 3$ for any $e \in \mathcal{F}(S_2) \cap \mathcal{F}(S_3)$: $\mathcal{F}(S_2) \cap \mathcal{F}(S_3)$ only contains cube edges of dimensions $2n - 1$. From Figure 19, the edge congestion contributed by $\mathcal{F}(S_2)$ is one. By condition 2 of the induction hypotheses for $k = n$, the same edges have a congestion of at most 2. Hence, the total congestion of these edges is at most 3.
- *cong*$(e) \leq 3$ for any $e \in \mathcal{F}(S_1) - \mathcal{F}(S_3)$: It is obvious from Figure 18.
- *cong*$(e) \leq 3$ for any $e \in \mathcal{F}(S_2) - \mathcal{F}(S_3)$: The edge congestion contributed by $\mathcal{F}(S_2)$ is one except that the edges in dimension $2n + 1$ in Figure 19 is two.
- *cong*$(e) \leq 3$ for any $e \in \mathcal{F}(S_3) - \mathcal{F}(S_1) - \mathcal{F}(S_2)$: By condition 1, the congestion is at most 3 for cube edges which do not have any new hyper-pyramid edges mapped onto them.

Therefore, the congestion of $f_{n+1}$ is at most 3 and satisfies condition 1 of the hypotheses. ∎

### 3.4.2 Active-degree and node-load

The active-degree of the embedding function is $2k+1$. For the node-load, we consider the following facts:

1. For $k=1$, the maximum $\beta_i$ for a node $i$ being the spare root, the large root, one of the small roots, and one of the large nodes at level one are 0, 4, 2 and 7, respectively.
2. A spare root after an induction step is a spare root before the induction step. The $\beta_i$ for the node does not increase during the induction step.
3. A large root after an induction step is a spare root before the induction step. The $\beta_i$ for the node increases at most by 4 during the induction step.
4. A small root after an induction step is a spare root before the induction step. The $\beta_i$ for the node increases at most by 6 during the induction step.
5. A large (small) node at level one after an induction step is a large (small) root before the induction step. The $\beta_i$ for the node increases at most by 7.
6. A large node at level $i$, $i>1$, after an induction step is a large node at level $i-1$ before the induction step. The $\beta_i$ for the node increases by 2.
7. A small node at level $i$, $i>1$, after an induction step is a small node at level $i-1$ before the induction step. The $\beta_i$ for the node can be arranged such that it increases by 2 and 4 alternately.

By these facts, one can prove inductively that the node-load

$$\beta = \begin{cases} 3k+4 & k \text{ is odd,} \\ 3k+5, & k \text{ is even.} \end{cases}$$

It is possible to have a node-load of $O(2k)$, if the dilation is relaxed to 4 and congestion to 5. Specifically, we follow the path assignments described in the proof of Lemma 10 without exchange of $\hat{P}_2$ and $\hat{P}_3$, i.e., $\gamma=0$. The proof is based on facts similar to the ones above except that $\beta_i$ in items 4, 5 and 7 above become 4, 9 and 2, respectively, after the induction step. The node-load is

$$\beta = \begin{cases} 7 & k=1, \\ 2k+9, & k \geq 2. \end{cases}$$

## 3.5 Remarks

When we incorporate $d=1$ into Theorem 1, the theorem becomes "A $\hat{P}(k,d)$ hyper-pyramid can be embedded in an $H_{kd+1}$ Boolean cube with dilation $\max(d,2)$". As a result, a $\hat{P}(k,1)$ hyper-pyramid can be embedded in an $H_{k+1}$ Boolean cube with dilation 2. Figure 20 shows a $\hat{P}(3,1)$ hyper-pyramid. Since an X-tree [11] of $k$ levels is a subgraph of the $\hat{P}(k,1)$ hyper-pyramid, an X-tree can be embedded in a Boolean cube with expansion $< 2$ and dilation 2. Figure 21 shows a 3-level X-tree.

Our result degenerates to that "a complete binary tree can be embedded in a hypercube with expansion $\approx 1$ and dilation 2", since a $\hat{P}(k,1)$ hyper-pyramid contains a complete binary tree as a subgraph. This result is not new and was first discovered by Nebeský [22] and rediscovered in [35], [3] and [10] independently. All embeddings except the one in [35] also guarantee that only one of the tree edges is of dilation 2. Our method is the same as that of [35] in which for every non-leaf node the edge to the left child is of dilation 1 and the edge to the right child is of dilation 2. However, in our embedding and the embedding in [35], all nodes at the same level forms a subcube and therefore has additional adjacencies, Figure 20. Our embedding and the embedding in [35] are equivalent to labeling a complete binary tree according to an inorder traversal [21] with a starting index of 0 or 1. Such an embedding was also used in [18,17].

Notice that a dilation 2 embedding of an X-tree can also be obtained by an inorder traversal by interpreting the label as a binary-reflected Gray code [28], as observed by Bhatt [1], Figure 22. (This is due to the property that two binary-reflected Gray codes with a power of two difference in their addresses is at most Hamming distance 2 apart [28].) However, the number of edges with dilation 2 is higher for such an embedding than for our embedding.

When the cube connections at level $i$ are ignored for $0 \leq i \leq k$ then the $\hat{P}(k,d)$ hyper-pyramid becomes a $k$-level complete $(2^d)$-ary tree. As a corollary of Theorem 1, a $k$-level complete $n$-ary tree can be embedded in a Boolean cube with dilation $\max(2, \lceil \log_2 n \rceil)$ and expansion $2^{k\lceil \log_2 n \rceil + 1} / \frac{n^{k+1}-1}{n-1}$. The expansion is less than two when $n$ is a power of two. The previous result by Wu [35] 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 $2^{\lceil \log_2 n \rceil} - 2$ $(k-1)$-level complete $n$-ary trees can be embedded in a $k\lceil \log_2 n \rceil + 1$ dimensional Boolean cube with dilation $\lceil \log_2 n \rceil + 1$. The expansion is approximately one when $n$ is a power of two.

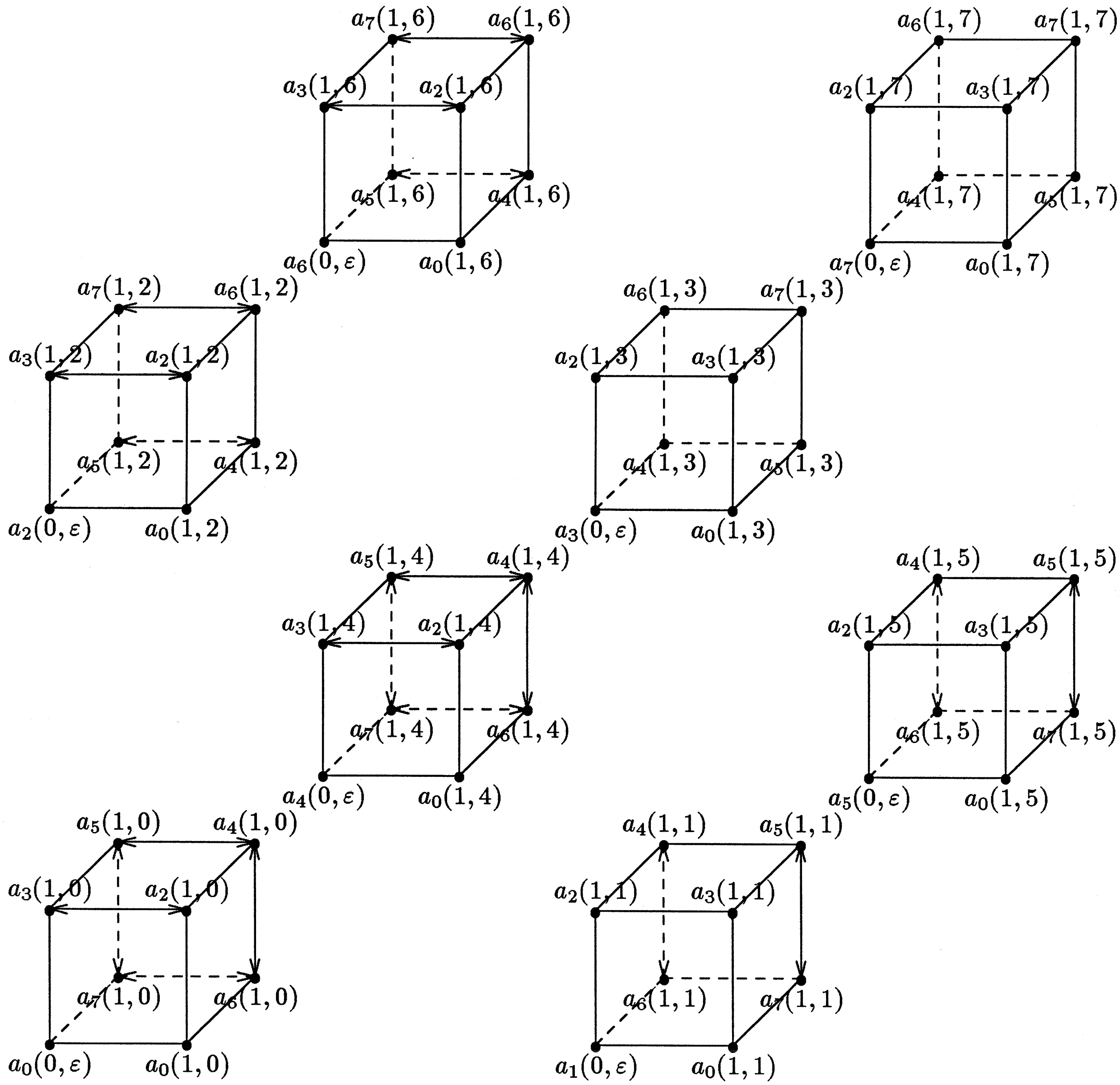



Figure 16: The "improved" embedding by performing an exchange described by $\gamma$ of the induction step for $d = 3$ with dilation 4.

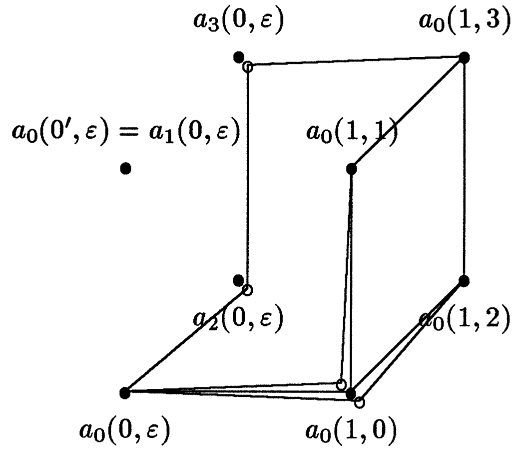



Figure 17: The basis, $k = 1$, satisfies the induction hypotheses.

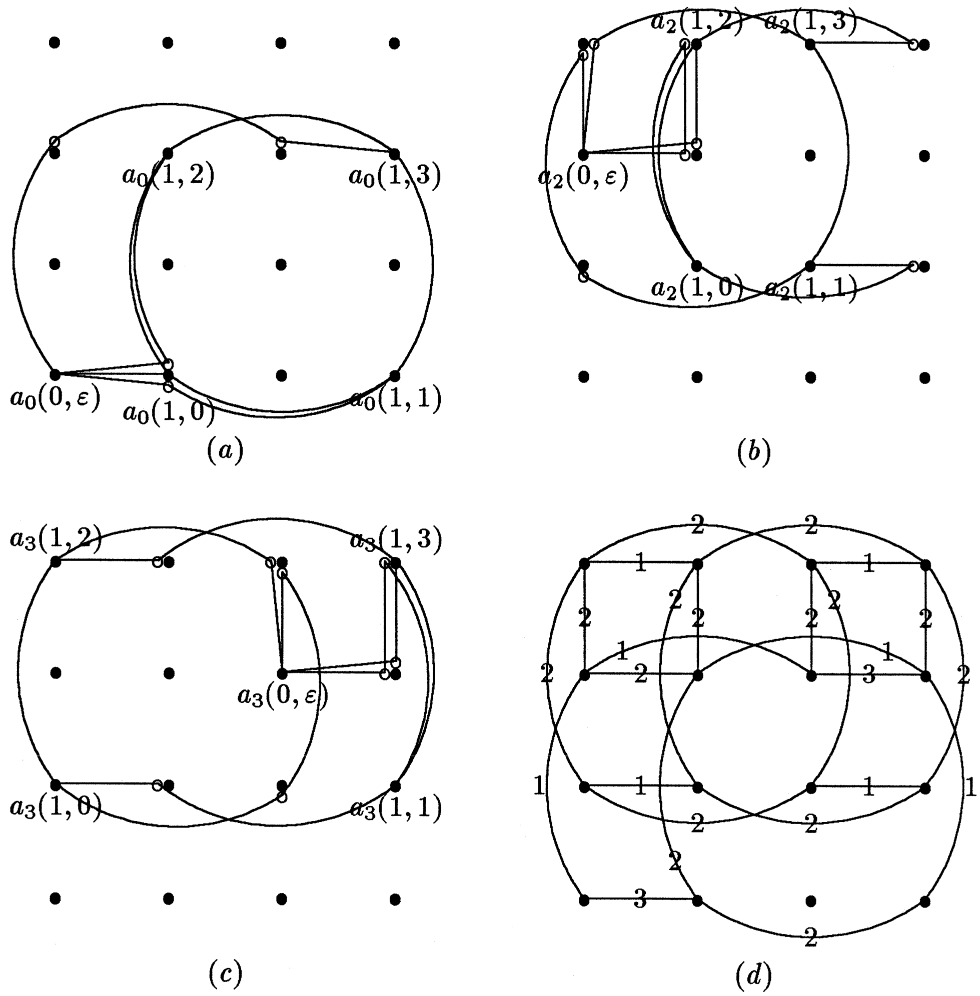



Figure 18: The path assignment of hyper-pyramid edges in $S_1$.

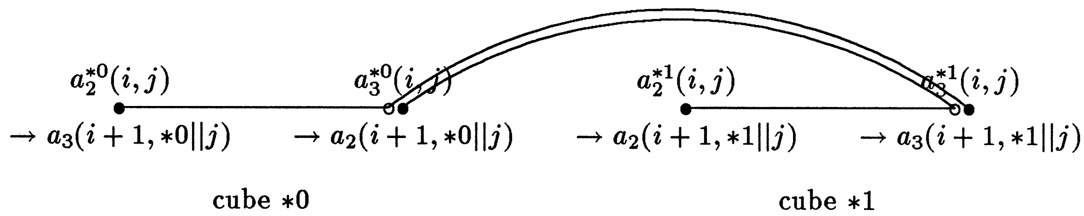



Figure 19: The path assignment for hyper-pyramid edges with dilation 2 in $S_2$. In the figure, $* = 0$ or 1.

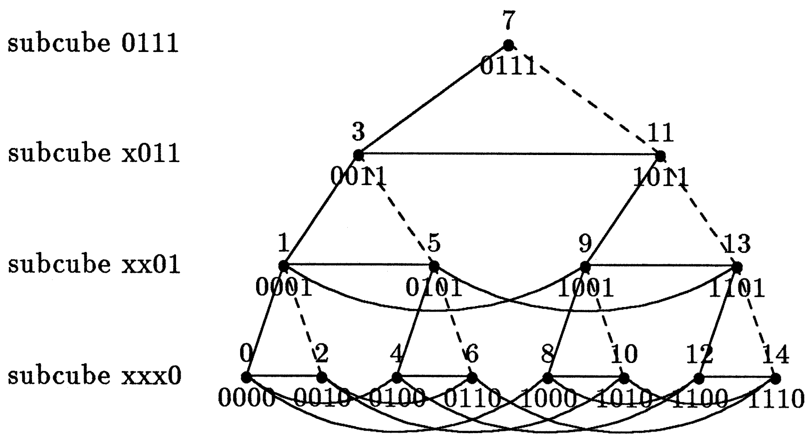



Figure 20: The topology of a $\hat{P}(3,1)$ hyper-pyramid. The dashed lines represent edges of dilation 2.

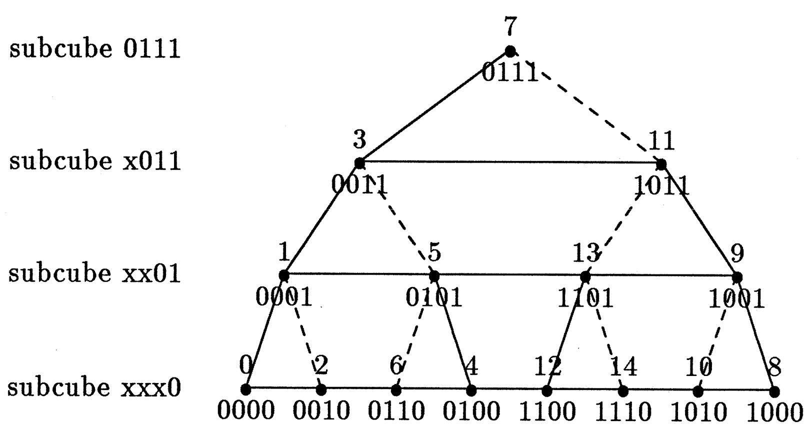



Figure 21: The topology of a 3-level X-tree. The dashed lines represent edges of dilation 2.

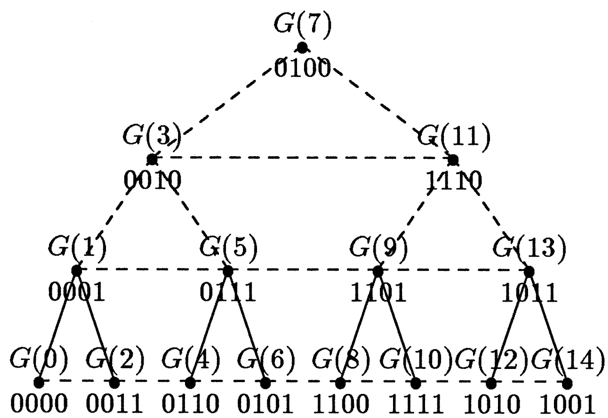



Figure 22: The topology of a 3-level X-tree. The dashed lines represent edges of dilation 2.

# 4 Summary

We have given embeddings of pyramids in hypercubes with minimal expansion, dilation 2 and congestion 2. We also give minimal expansion and dilation $d$ embeddings of hyper-pyramids, i.e., pyramids where a node has $2^d$ children interconnected as Boolean cubes. The congestion is bounded from below by $1 + \lceil \frac{2^d - d}{kd+1} \rceil$ and from above by $\frac{2^{d+1}}{d+1}$.

The expansion is asymptotically 1.5 for the embedding of a pyramid $P(k, 2, 2)$, and 2 for the embedding of a hyper-pyramid $\hat{P}(k, d)$. In the first case about $\frac{1}{3}$ of the cube nodes are unused, and in the second about half of them are unused. By embedding two pyramids of height $k - 1$ together with a pyramid of height $k$ the expansion becomes approximately one. Lai and White [24] gave such an embedding with dilation 3 and congestion 6. We improve it to dilation 3 and congestion 3. The same expansion can be obtained for hyper-pyramids by embedding $2^d - 2$ hyper-pyramids of height $k - 1$ with a hyper-pyramid of height $k$. One of our embeddings has dilation $d + 1$ and its congestion is bounded from above by $O(2^d)$. The other embedding has a congestion of $\lceil \frac{2^d}{d} \rceil + \frac{2^{d+1}}{d+2} + 1$ for a dilation of $2d$.

It follows from the hyper-pyramid embeddings that a $P(k, 2^i, 2^{d-i})$ pyramid can be embedded in a hypercube with minimal expansion, dilation $d$, and a congestion of at most $\frac{2^{d+1}}{d+1}$. A pyramid and $2^d - 2$ smaller pyramids of $P(k, 2^i, 2^{d-i})$ (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 $\max(2, \lceil \log_2 n \rceil)$. Previously best-known embedding has dilation $2\lceil \log_2 n \rceil$ [35]. Our results also provide embeddings of degenerate hyper-pyramids such as complete binary trees and X-trees with minimal expansion and dilation 2.

In the analysis we also determine the maximum number of host graph edges that are used for any host node, the *active-degree*, and the maximum number of messages a node has to transmit, the *node-load*. In the embeddings we give, all edges are used for some host nodes, i.e. the active-degree is $kd + 1$ for a $P(k, d)$ hyper-pyramid. The node-load is of order $O(\frac{2^d}{d} + kd)$. Table 3

| Embedding | Active-degree | Node-load | Dilation | Congestion |
|---|---|---|---|---|
| $\hat{P}(k,d)$ | $kd+1$ | $(1+\frac{4}{d+1})2^d+(k-1)d$ | $d$ | $\leq \frac{2^{d+1}}{d+1}$ |
| $\hat{P}(k,2)$ | $2k+1$ | $2k+5$ | $2$ | $2$ |
| $\hat{P}(k,d)$ and $(2^d-2)\ \hat{P}(k-1,d)$ | $kd+1$ | $O(d2^d)+O(kd)$ | $d+1$ | $O(2^d)$ |
| | | $(3+\frac{4}{d+2})2^d+(k-3)d,\ k\geq 3$ | $2d$ | $\leq \lceil \frac{2^d}{d} \rceil + \frac{2^{d+1}}{d+2}+1$ |
| $\hat{P}(k,2)$ and $2\ \hat{P}(k-1,2)$ | $2k+1$ | $3k+5$ | $3$ | $3$ |
| | | $2k+9$ | $4$ | $5$ |

Table 3: Active-degree and node-load for hyper-pyramid embedding.

summarizes the results.

# Appendix

## A Proof of the lemma used in Theorem 3

In the following, $\gamma(\ell,x)$ is defined in Section 3.3.1 as follows: $\gamma(\ell,x)=(\gamma_{d-1}\gamma_{d-2}\ldots\gamma_0)$, where for all $0\leq m\leq d-1$: $\gamma_m=1$, if $\ell_m=0$ and $(x_{d-1}x_{d-2}\ldots x_{m+1})\neq 0$; and $\gamma_m=0$, otherwise. (So, $\gamma_{d-1}=0$, by definition.)

**Lemma 11** $\gamma(\ell,x)=\gamma(\ell,x\oplus\gamma(\ell,x))$.

**Proof:** Let $y=\gamma(\ell,x)$ and $\gamma'=\gamma(\ell,x\oplus y)$. Then, we prove that $\gamma'_m=y_m$ for all $0\leq m\leq d-1$. Let $x'=x\oplus y$.

- If $\ell_m=1$, then $y_m=\gamma'_m=0$.
- If $\ell_m=0$:
  - $(x_{d-1}x_{d-2}\ldots x_{m+1})=0$: Then, $y_m=0$ and $(y_{d-1}y_{d-2}\ldots y_{m+1})=0$. So, $(x'_{d-1}x'_{d-2}\ldots x'_{m+1})=0$ too, i.e., $\gamma'_m=0$. Therefore, $y_m=\gamma'_m$.
  - $(x_{d-1}x_{d-2}\ldots x_{m+1})\neq 0$: Then, $y_m=1$. Let $x_r$ be the leading non-zero bit, $m+1\leq r\leq d-1$. Then, $y_r=0$, and $x'_r=x_r\oplus y_r=1$, i.e., $(x'_{d-1}x'_{d-2}\ldots x'_{m+1})\neq 0$. So, $\gamma'_m=1$. Therefore, $y_m=\gamma'_m$. ∎

**Lemma 12** $||x\oplus\gamma(\ell,x)||+||x\oplus\ell||\leq d+1$ *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 1-bit of $x$ ($r = -1$, if $x = 0$). Consider any $m$ such that $x_m \oplus \ell_m = 1$

- $m < r$: If $x_m = 0$ then $\ell_m = 1$, $\gamma_m = 0$. If $x_m = 1$ then $\ell_m = 0$, $\gamma_m = 1$. Both have $(x_m \oplus \gamma_m) + (x_m \oplus \ell_m) = 1$.
- $m = r$: $x_m = x_r = 1$, so $\ell_m = 0$ and $\gamma_m = 0$ (since $x_{d-1}x_{d-2}\ldots x_{m+1} = 0$). We have $(x_m \oplus \gamma_m) + (x_m \oplus \ell_m) = 2$.
- $m > r$: $x_m = 0$ since $x_r$ is the leading 1-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 it sums up to 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,

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

**Acknowledgement** We thank David Greenberg for his comments on a draft of this paper and Nancy Elder for proofreading the manuscript. The generous support by the Office of Naval Research under Contract No. N00014-86-K-0310 is gratefully acknowledged.

# References

[1] Sandeep N. Bhatt. 1987. Personal communication.

[2] Sandeep N. Bhatt, F.R.K. Chung, F. Tom Leighton, and Arnold L. Rosenberg. Optimal simulations of tree machines. In *Proc. 27th IEEE Symp. Foundations Comput. Sci.*, pages 274–282, IEEE Computer Society, 1986.

[3] 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 Univ., New Haven, CT, December 1985.

[4] Sandeep N. Bhatt and Charles E. Leiserson. *How to Assemble Tree Machines*, pages 95–114. Volume 2, JAI Press Inc., 1984.

[5] V. Cantoni and S. Levialdi, editors. *Pyramidal Systems for Computer Vision*, Spring-Verlag Berlin Heidelberg, 1986.

[6] M.Y. Chan. *Dilation-2 Embeddings of Grids into Hypercubes.* Technical Report UTDCS 1-88, Computer Science Dept., University of Texas at Dallas, 1988.

[7] M.Y. Chan and F.Y.L. Chin. *On Embedding Rectangular Grids.* Technical Report TR-B2-87, Center of Computer Studies and Applications, University of Hong Kong, February 1987. to appear in IEEE Trans. Computers.

[8] Tony F. Chan and Yousef Saad. Multigrid algorithms on the hypercube multiprocessor. *IEEE Trans. Computers*, 35(11):969–977, November 1986.

[9] J.H. Chang, O.H. Ibarra, T.C. Pong, and S.M. Sohn. Two-dimensional convolution on a pyramid computer. In *1987 International Conf. on Parallel Processing*, pages 780–782, IEEE Computer Society, 1987.

[10] Sanjay R. Deshpande and Roy M. Jenevin. Scaleability of a binary tree on a hypercube. In *1986 International Conference on Parallel Processing*, pages 661–668, IEEE Computer Society, 1986.

[11] A.M. Despain and D.A. Patterson. X-tree – a tree structured multiprocessor architecture. In *5th Symp. on Computer Architecture*, pages 144–151, 1978.

[12] David S. Greenberg. *Minimum Expansion Embeddings of Meshes in Hypercubes.* Technical Report YALEU/DCS/RR-535, Dept. of Computer Science, Yale Univ., New Haven, CT, August 1987.

[13] Ivan Havel and Petr Liebl. Embedding the polytomic tree into the $n$-cube. *Časopis pro Pěstování matematiky*, 98:307–314, 1973.

[14] Ching-Tien Ho and S. Lennart Johnsson. Distributed routing algorithms for broadcasting and personalized communication in hypercubes. In *1986 International Conf. on Parallel Processing*, pages 640–648, IEEE Computer Society, 1986. Tech. report YALEU/DCS/RR-483, May 1986.

[15] Ching-Tien Ho and S. Lennart Johnsson. On the embedding of arbitrary meshes in Boolean cubes with expansion two dilation two. In *1987 International Conf. on Parallel Processing*, pages 188–191, IEEE Computer Society, 1987. Report YALEU/DCS/RR-576, April 1987.

[16] Ching-Tien Ho and S. Lennart Johnsson. *Spanning balanced trees in Boolean cubes.* Technical Report YALEU/DCS/RR-508, Dept. of Computer Science, Yale Univ., New Haven, CT, January 1987. To appear in SIAM J. Sci. Statist. Comput.

[17] S. Lennart Johnsson. Communication efficient basic linear algebra computations on hypercube architectures. *J. Parallel Distributed Comput.*, 4(2):133–172, April 1987. (Tech. Rep. YALEU/DCS/RR-361, Yale Univ., New Haven, CT, January 1985).

[18] S. Lennart Johnsson. *Odd-even cyclic reduction on ensemble architectures and the solution tridiagonal systems of equations.* Technical Report YALE/DCS/RR-339, Dept. of Computer Science, Yale Univ., October 1984.

[19] S. Lennart Johnsson and Ching-Tien Ho. *Spanning graphs for optimum broadcasting and personalized communication in hypercubes.* Technical Report YALEU/DCS/RR-500, Dept. of Computer Science, Yale Univ., New Haven, CT, November 1986. Revised November 1987, YALEU/DCS/RR-610. To appear in IEEE Trans. Computers.

[20] S. Lennart Johnsson and Peggy Li. *Solutionset for AMA/CS 146.* Technical Report 5085:DF:83, California Institute of Technology, May 1983.

[21] Donald E. Knuth. *The Art of Computer Programming, Vol. 1: Fundamental Algorithms.* Addison-Wesley, 1968.

[22] Ladislav Nebeský. On cubes and dichotomic trees. *Časopis pro Pěstování matematiky*, 99:164–167, 1974.

[23] Ten-Hwang Lai and William White. *Embedding Pyramids in Hypercubes.* Technical Report, Dept. of Computer and Information Science, Ohio State Univ., November 1987.

[24] Ten-Hwang Lai and William White. *Mapping Multiple Pyramids into Hypercubes Using Unit Expansion.* Technical Report, Dept. of Computer and Information Science, Ohio State Univ., January 1988.

[25] F. Tom Leighton, F. Makedon, and I.H. Sudburough. 1988. Personal communication.

[26] Marilynn Livingston and Quentin F. Stout. Embeddings in hypercubes. In *Proceedings Sixth International Conference on Mathematical Modeling*, August 1987.

[27] Russ Miller and Quentin F. Stout. Data movement techniques for the pyramid computer. *SIAM J. Comput.*, 16(1):38–60, February 1987.

[28] E M. Reingold, J Nievergelt, and N Deo. *Combinatorial Algorithms.* Prentice-Hall, Englewood Cliffs. NJ, 1977.

[29] Arnold L. Rosenberg and Larry Snyder. Bounds on the costs of data encodings. *Mathematical Systems Theory*, 12:9–39, 1978.

[30] Yousef Saad and Martin H. Schultz. *Topological properties of hypercubes.* Technical Report YALEU/DCS/RR-389, Dept. of Computer Science, Yale Univ., New Haven, CT, June 1985.

[31] Quentin F. Stout. Hypercubes and pyramids. In V. Cantoni and S. Levialdi, editors, *Pyramidal Systems for Computer Vision*, Spring-Verlag Berlin Heidelberg, 1986.

[32] Quentin F. Stout. Sorting, merging, selecting, and filtering on tree and pyramid machines. In *1983 International Conf. on Parallel Processing*, pages 214–221, IEEE Computer Society, 1983.

[33] Leonard Uhr. Parallel, hierarchical software/hardware pyramid architectures. In V. Cantoni and S. Levialdi, editors, *Pyramidal Systems for Computer Vision*, Spring-Verlag Berlin Heidelberg, 1986.

[34] Alan S. Wagner. *Embedding Trees in the Hypercube.* PhD thesis, University of Toronto, 1987.

[35] Angela Y. Wu. Embedding of tree networks in hypercubes. *Journal of Parallel and Distributed Computing*, 2(3):238–249, 1985.