

Purdue University  
**Purdue e-Pubs**

---

Department of Computer Science Technical  
Reports

Department of Computer Science

---

1989

## Multiple Network Embeddings into Hypercubes

Ajay K. Gupta

Susanne E. Hambrusch  
*Purdue University, seh@cs.purdue.edu*

Report Number:  
89-878

---

Gupta, Ajay K. and Hambrusch, Susanne E., "Multiple Network Embeddings into Hypercubes" (1989).  
*Department of Computer Science Technical Reports*. Paper 746.  
<https://docs.lib.purdue.edu/cstech/746>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries.  
Please contact [epubs@purdue.edu](mailto:epubs@purdue.edu) for additional information.

MULTIPLE NETWORK EMBEDDINGS INTO HYPERCUBES

Ajay K. Gupta  
Susanne E. Hambrusch

CSD-TR-878  
April 1989

## **Multiple Network Embeddings into Hypercubes**

Ajay K. Gupta  
Susanne E. Hambrusch

CC-89-8 (August 1989)

Purdue University  
West Lafayette, Indiana 47907

This research was supported in part by the Office of Naval Research under University Research Initiative grant number N00014-86-K-0689.

# Multiple Network Embeddings into Hypercubes

*Ajay K. Gupta*

Department of Computer Sciences  
Purdue University  
West Lafayette, IN 47907

*Susanne E. Hambrusch*

International Computer Science Institute  
Berkeley, CA 94704

and

Department of Computer Sciences  
Purdue University  
West Lafayette, IN 47907

April 1989

This work was supported by the Office of Naval Research under Contracts  
N00014-84-K-0502 and N00014-86-K-0689, and by the National Science  
Foundation under Grant MIP-87-15652.

### Abstract

In this paper we study the problem of how to efficiently embed  $r$  interconnection networks  $G_0, \dots, G_{r-1}$ ,  $r \leq k$ , into a  $k$ -dimensional hypercube  $H$  so that every node of the hypercube is assigned at most  $r$  nodes all of which belong to different  $G_i$ 's. When each  $G_i$  is a complete binary tree or a leap tree of  $2^k - 1$  nodes, we describe an embedding achieving a dilation of 2 and a load of 5 and 6, respectively. For the cases when each  $G_i$  is a linear array or a 2-dimensional mesh of  $2^k$  nodes, we describe embeddings that achieve a dilation of 1 and an optimal load of 2 and 4, respectively. Using these embeddings, we also show that  $r_1$  complete binary trees,  $r_2$  leap trees,  $r_3$  linear arrays, and  $r_4$  meshes can simultaneously be embedded into  $H$  with constant dilation and load,  $\sum_{i=1}^4 r_i \leq k$ .

# 1 Introduction

The problem of embedding a single  $m$ -processor source network  $G$  into an  $n$ -processor host network  $H$  is an important problem in parallel processing and it has been studied extensively [3,5,8,10,11,13,16]. An embedding of a single source network  $G$  into a host  $H$  does not always make use of all the resources available in  $H$ . If the computational environment allows simultaneous use of the host by different interconnection networks, the problem of how to efficiently embed a number of networks arises. In this paper we consider the problem of embedding multiple networks when the host network is a hypercube. Hypercube architectures have been built successfully [1,12] and efficient embeddings of many constant-degree networks  $G$  into  $H$  utilize only  $\Theta(n)$  of the  $\Theta(n \log n)$  edges of  $H$  [6,7,8,16]. We show that up to  $\Theta(\log n)$  instances of frequently used constant-degree networks can simultaneously be embedded into  $H$  without a significant increase in dilation and load compared to the embedding of a single source network.

Let  $n = 2^k$  and let  $G_0, G_1, \dots, G_{r-1}$  be the source networks to be embedded into a  $k$ -dimensional hypercube  $H$ ,  $r \leq k$ . We consider embeddings for complete binary trees, leap trees, linear arrays, and meshes. In our embeddings a processor of  $H$  is assigned at most  $r$  processors and no processor of  $H$  has two processors from the same  $G_i$  assigned to it. We show that  $\log n$  complete binary trees, each with  $n - 1$  processors, can be embedded into  $H$  so that the dilation is 2 and the load is 5. This result also gives an embedding of  $\log n$  leap trees with a dilation of 2 and a load of 6. We also present an optimal embedding of  $\log n$   $n$ -processor linear arrays into  $H$  that achieves a dilation of 1 and a load of 2, and an embedding of  $\log n$

$n$ -processor meshes into  $H$  that achieves a dilation of 1 and a load of 4. Using these embeddings we show that  $r_1$  complete binary trees,  $r_2$  leap trees,  $r_3$  linear arrays, and  $r_4$  meshes can be simultaneously embedded into  $H$  with a dilation of at most 2 and a load of at most 12, for  $\sum_{i=1}^4 r_i \leq k$ .

An obvious solution for embedding  $G_0, G_1, \dots, G_{r-1}$  into  $H$  so that all the  $r$  processors assigned to a processor of  $H$  come from different  $G_i$ 's is to use the same embedding for every  $G_i$ . Doing so always results in a load of  $\Theta(r)$ . For example, one complete binary tree can be optimally embedded into  $H$  with a dilation of 2 and a load of 1 [7]. Using this embedding  $r$  times gives a load of  $r$  on  $n - 2$  edges of  $H$ , while  $\frac{n}{2} \log n - n + 2$  edges of  $H$  carry a load of zero. In this paper we describe embeddings that distribute the load among the edges of  $H$  so that it is reduced to a constant and thereby utilize all the edges of  $H$  evenly within a constant factor. We next give definitions and notations used throughout this paper.

An embedding  $\langle f, g \rangle$  of  $G$  into  $H$  is defined by a surjective mapping  $f$  from the processors of  $G$  to the processors of  $H$  together with a mapping  $g$  that maps every edge  $e = (v, w)$  of  $G$  onto a path  $g(e)$  connecting  $f(v)$  and  $f(w)$ . We refer to  $f$  as the assignment. Two fundamental cost measures of an embedding are the dilation and load [2,6,13,15]. The *dilation*  $\delta$  is defined as the maximum distance in  $H$  between two adjacent processors in  $G$ , and the *load*  $\lambda$  is defined as the maximum number of paths containing an edge in  $H$ , where every path represents an edge in  $G$ .

Every processor in the  $k$ -dimensional hypercube  $H$  is labeled as  $b_0 b_1 \dots b_{k-1}$ , where  $b_s = 0, 1$  for  $0 \leq s \leq k - 1$ . A processor with label  $b_0 b_1 \dots b_{k-1}$  is connected to  $k$  processors having labels  $b_0 b_1 \dots \bar{b}_s \dots b_{k-1}$ , for  $0 \leq s \leq k - 1$ .

We call an edge  $(v_1, v_2)$  of  $H$  to be an *edge of dimension  $s$* , if  $v_1$  and  $v_2$  differ in bit position  $s$ , i.e.,  $v_1 = b_0b_1 \dots b_s \dots b_{k-1}$  and  $v_2 = b_0b_1 \dots \bar{b}_s \dots b_{k-1}$ . For clarity reasons, henceforth we will refer to the processors of source networks as PEs and to the processors of  $H$  as nodes.

In Section 2 we describe the embedding of  $(n - 1)$ -processor complete binary trees  $T_0, T_1, \dots, T_{k-1}$  into hypercube  $H$  and also show how this embedding gives an embedding for leap trees. In order to achieve a load of 5 our embedding assigns the roots of the  $T_i$ 's to different nodes in  $H$ , each one having a unique "mark-position". We use the dimensions of the hypercube in a "cyclic" order, i.e., when an edge of  $T_0$  is mapped to an edge of dimension  $s$ , then the corresponding edges of  $T_1, T_2, \dots, T_{k-1}$  are mapped to edges of dimension  $s + 1, s + 2, \dots, k - 1, 0, \dots, s - 1$ . A similar strategy is used to achieve a constant load in the embeddings of the linear arrays and the meshes that are described in Section 3. Section 4 describes how to use the embeddings of Sections 2 and 3 to get efficient embeddings when the source networks are not of the same type.

## 2 Embedding $k$ Complete Binary Trees

In this section we consider the problem of embedding  $k$   $(n - 1)$ -PE complete binary trees  $T_0, T_1, T_2, \dots, T_{k-1}$  into  $H$  when the  $k$  PEs assigned to a node of  $H$  have to come from different trees. As stated in the previous section, a brute-force solution is to use the same embedding of one tree into  $H$   $k$  times. Doing so results in a load of  $k$  over  $n - 2$  edges of  $H$ , while  $\frac{n}{2} \log n - n + 2$  edges have zero load. Our embedding achieves a dilation of 2 and a load of 5.



We start by describing a well-known embedding of a complete binary tree  $T$  into  $H$  that is based on the inorder numbering. Let  $u$  be a PE of  $T$  at level  $\alpha$ ,  $0 \leq \alpha \leq k - 1$ , and let  $in(u)$  be its inorder number, where the inorder number of the leftmost leaf of  $T$  is 0. PE  $u$  is assigned to node  $v$  of  $H$  if and only if  $v = in(u)$ . Let  $r(u)$ ,  $l(u)$ , and  $s(u)$  be the right and left child, and the sibling of  $u$ , respectively. Then, the assignment maps the edge  $(u, l(u))$  onto an edge of dimension  $\alpha + 1$  in  $H$ . The edge  $(u, r(u))$  is mapped onto a path of length 2 consisting of an edge of dimension  $\alpha + 1$  followed by one of dimension  $\alpha$ . This embedding achieves a dilation of 2 and a load of 2. When using the embedding  $k$  times, we get a load of  $2k$ . The total load of all the edges incident to a node is at most  $5k$  and  $k - 3$  edges have a load of 0. We next describe the embedding that reduces the load to 5 by distributing the total load at every node evenly among the  $k$  edges incident to that node.

The root of tree  $T_i$  is assigned to node  $1^i 01^{k-i-1}$  and we refer to the 0 in position  $i$  as the *mark-bit* of  $T_i$ . Let  $u$  be a PE of  $T_i$  on level  $\alpha$  assigned to node  $v$  of  $H$ ,  $0 \leq \alpha \leq k - 2$ . Then  $l(u)$  is assigned to node  $v'$  adjacent to  $v$  so that  $(v, v')$  is an edge of dimension  $\alpha + i + 2$ .<sup>1</sup> PE  $r(u)$  is assigned to node  $v''$  adjacent to  $v'$  so that  $(v', v'')$  is an edge of dimension  $\alpha + i + 1$ . The resulting embedding of  $T_0, T_1, \dots, T_{k-1}$  has the following two properties that are crucial for achieving a load of 5.

1. The mark-bit is changed only when assigning the leaves of the  $T_i$ .
2. If dimension  $s$  is used for embedding an edge of  $T_i$ , then the embedding of tree  $T_{i+1}$  uses dimension  $s + 1$  for the corresponding edge.

---

<sup>1</sup> Throughout, additions and subtractions are computed using *modulo k* operation.

In order to have edges being mapped to edges (instead of paths), we switch from the embedding of complete binary trees to the embedding of sibling trees. We call a tree  $T'_i$  to be the *sibling tree* of  $T_i$  when every edge  $(u, r(u))$  in  $T_i$  is replaced by a second edge  $(u, l(u))$  and an edge  $(l(u), r(u))$ . We refer to the two edges from  $u$  to  $l(u)$  in  $T'_i$  as a *double-edge* and to the edge from  $l(u)$  to  $s(l(u))$  as a *single-edge*. Obviously, the embedding of the  $T_i$ 's given above also embeds the  $T'_i$ 's. Figure 1 shows the embedding of  $T'_1$  into  $H$  when  $k = 4$ . The numbers on the edges of  $T'_i$  indicate the dimension of the hypercube used by that edge. When an edge  $e_1$  of  $T'_i$  is assigned to an edge  $e_2$  of  $H$ , we say that edge  $e_2$  is *used* by edge  $e_1$  or by tree  $T'_i$ . We next show that the embedding of  $T'_0, T'_1, \dots, T'_{k-1}$  has a load of 5 by showing that every edge in  $H$  is used by at most two double-edges and one single-edge.

To simplify the notation we henceforth refer to the labels of the nodes that have PEs of  $T'_i$  assigned as  $b_i b_{i+1} \dots b_{k-1} b_0 \dots b_{i-1}$  instead of  $b_0 b_1 \dots b_{k-1}$ . Using this notation, the root  $r_i$  is assigned to node  $0_i 1^{k-1}$  and the two PEs on level 1 are assigned to nodes<sup>2</sup>  $0_i * 0 1^{k-3}$ . The subscript  $i$  in the label indicates the bit position  $i$ .

Let  $(v_1, v_2)$  be an edge of  $H$ . We show that  $(v_1, v_2)$  is used by at most two double-edges and one single-edge in our embedding. To show this, we partition the edges of  $T'_i$  in 6 sets, namely sets R1, R2, S1, S2, L1, and L2, and prove 3 lemmas that characterize the edges of the hypercube depending on how the  $T'_i$ 's use them. We omit an indexing on  $i$  for the sets since it will be clear from the context to which tree a specified set belongs. The

---

<sup>2</sup> \* in the label denotes a wild card character indicating 0 and 1.

edges in R1, S1, and L1 are the double-edges and the edges in R2, S2, and L2 are the single-edges in  $T'_i$ . The six sets are defined as follows.

$$\mathbf{R1} = \{(0_i 11_{i+2} 1^{k-3}, 0_i 10_{i+2} 1^{k-3})\}.$$

Set R1 contains one edge, namely the double-edge that connects the root  $r_i$  of  $T'_i$  to the left child  $l(r_i)$ .

$$\mathbf{R2} = \{(0_i 1_{i+1} 01^{k-3}, 0_i 0_{i+1} 01^{k-3})\}.$$

Set R2 contains one edge, namely the single-edge that connects the left child  $l(r_i)$  of the root to  $s(l(r_i))$ .

$$\mathbf{S1} = \{(0_i *^\alpha 01_s 1^{k-\alpha-3}, 0_i *^\alpha 00_s 1^{k-\alpha-3}) \mid s = i+3, \dots, k-1, 0, \dots, i-1 \text{ and } \alpha = s - i - 2\}.$$

Set S1 contains the  $(\frac{n}{4} - 2)$  double-edges that connect a PE  $u$  on level  $\alpha$  of  $T'_i$  to PE  $l(u)$  on level  $\alpha + 1$ ,  $1 \leq \alpha \leq k - 3$ .

$$\mathbf{S2} = \{(0_i *^\alpha 0_s 01^{k-\alpha-3}, 0_i *^\alpha 1_s 01^{k-\alpha-3}) \mid s = i+2, \dots, k-1, 0, \dots, i-2 \text{ and } \alpha = s - i - 1\}.$$

Set S2 contains the  $(\frac{n}{4} - 2)$  single-edges that connect a PE  $u$  on level  $\alpha + 1$  of  $T'_i$  to PE  $s(u)$  on level  $\alpha + 1$ ,  $1 \leq \alpha \leq k - 3$ .

$$\mathbf{L1} = \{(0_i *^{k-2} 0, 1_i *^{k-2} 0)\}.$$

Set L1 contains the  $\frac{n}{4}$  double-edges that connect a PE  $u$  on level  $k - 2$  of  $T'_i$  to  $l(u)$  on level  $k - 1$  (i.e.,  $l(u)$  is a leaf PE).

$$\mathbf{L2} = \{(1_i *^{k-2} 0_{i-1}, 1_i *^{k-2} 1_{i-1})\}.$$

Set L2 contains the  $\frac{n}{4}$  single-edges that connect a PE  $u$  on level  $k - 1$  of  $T'_i$  to  $s(u)$  on level  $k - 1$ , i.e., the edges belonging to L2 connect two leaf PEs.

Let  $s$  be an integer,  $0 \leq s \leq k - 1$ , and let  $H_s$  be the set of the  $\frac{n}{2}$  edges

of dimension  $s$  in  $H$ . Then  $T'_s$  and  $T'_{s+1}$  use the edges of  $H_s$  in the following way.

**Lemma 1:** *Set  $H_s$  can be partitioned into four sets  $H_s^1$ ,  $H_s^2$ ,  $H_s^3$ , and  $H_s^4$  with  $|H_s^l| = \frac{n}{8}$ ,  $1 \leq l \leq 4$ , such that*

- (a) *every edge in  $H_s^1$  is of the form  $(0_s 0 *^{k-3} 0, 1_s 0 *^{k-3} 0)$  and it is used by a double edge of  $T'_s$  and by a double-edge of  $T'_{s+1}$ .*
- (b) *every edge in  $H_s^2$  is of the form  $(0_s 1 *^{k-3} 0, 1_s 1 *^{k-3} 0)$  and it is used by a double edge of  $T'_s$  and by a single-edge of  $T'_{s+1}$ .*
- (c) *every edge in  $H_s^3$  is of the form  $(0_s 1 *^{k-3} 1, 1_s 1 *^{k-3} 1)$  and it is used by a single-edge of  $T'_{s+1}$  and it is not used by any edge of  $T'_s$ .*
- (d) *every edge in  $H_s^4$  is of the form  $(0_s 0 *^{k-3} 1, 1_s 0 *^{k-3} 1)$  and it is not used by any edge of  $T'_s$  or  $T'_{s+1}$ .*

**Proof:** The edges of  $T'_s$  that use edges in  $H_s$  are exactly the  $n/4$  edges in L1, which are double-edges of the form  $(0_s *^{k-2} 0, 1_s *^{k-2} 0)$ . Hence,  $n/8$  of these edges are in  $H_s^1$  and the other  $n/8$  edges are in  $H_s^2$ . The edges of  $T'_{s+1}$  that use edges in  $H_s$  are the  $n/8$  double-edges in S1 of the form  $(1_s 0 *^{k-3} 0, 0_s 0 *^{k-3} 0)$  and the  $n/4$  single-edges in L2 of the form  $(0_s 1 *^{k-2}, 1_s 1 *^{k-2})$ . The  $n/8$  double-edges in S1 of  $T'_{s+1}$  are in  $H_s^1$  and hence claim (a) follows. Half of the  $n/4$  single-edges in L2 of  $T'_{s+1}$  are in  $H_s^2$  and the other half are in  $H_s^3$ . No other edges of  $T'_{s+1}$  use edges in  $H_s^2$  or  $H_s^3$  and hence claims (b) and (c) follow. Since all the edges of dimension  $s$  that are used by either  $T'_s$  or  $T'_{s+1}$  belong to sets  $H_s^1$ ,  $H_s^2$ , or  $H_s^3$ , no edge in  $H_s^4$  is used by an edge of  $T'_s$  or  $T'_{s+1}$  and the lemma follows. ■

We next show that if an edge of  $T'_i$  in set S1 (resp. S2) uses an edge  $(v_1, v_2)$  of  $H$ , then no other edge in set S1 (resp. S2) belonging to some other  $T'_j$  uses  $(v_1, v_2)$ .

**Lemma 2:** *Let  $(v_1, v_2)$  be an edge of dimension  $s$  in  $H$ . Let  $(u_{i1}, u_{i2})$  be a double-edge in set S1 of  $T'_i$  and let  $(u_{j1}, u_{j2})$  be a double-edge in set S1 of  $T'_j$ , for  $i \neq j \neq s+1$  and  $0 \leq i, j \leq k-1$ . If  $(v_1, v_2)$  is used by  $(u_{i1}, u_{i2})$ , then it can not be used by  $(u_{j1}, u_{j2})$ .*

**Proof:** We know that  $(u_{i1}, u_{i2}) = (1_s 1^{k-\alpha-3} 0_i *^\alpha 0, 0_s 1^{k-\alpha-3} 0_i *^\alpha 0)$  and  $(u_{j1}, u_{j2}) = (1_s 1^{k-\beta-3} 0_j *^\beta 0, 0_s 1^{k-\beta-3} 0_j *^\beta 0)$ , for  $\alpha = s-i-2$  and  $\beta = s-j-2$ . Depending on whether  $\alpha < \beta$  or not, we distinguish two cases. Note that since  $i \neq j$ , we have  $\alpha \neq \beta$ .

**Case 1:**  $\alpha < \beta$ . In this case  $u_{i1} = 1_s 1^{k-\beta-3} 1 1^{\beta-\alpha-1} 0_i *^\alpha 0$  and  $u_{j1} = 1_s 1^{k-\beta-3} 0_j *^{\beta-\alpha-1} *^\alpha 0$ . Since  $b_j = 1$  in  $u_{i1}$  and  $b_j = 0$  in  $u_{j1}$ , if  $(u_{i1}, u_{i2})$  uses  $(v_1, v_2)$ , then  $(u_{j1}, u_{j2})$  can not use  $(v_1, v_2)$ .

**Case 2:**  $\alpha > \beta$ . In this case  $u_{i1} = 1_s 1^{k-\alpha-3} 0_i *^{\alpha-\beta-1} *^\beta 0$  and  $u_{j1} = 1_s 1^{k-\alpha-3} 1 1^{\alpha-\beta-1} 0_j *^\beta 0$ . Since  $b_i = 0$  in  $u_{i1}$  and  $b_i = 1$  in  $u_{j1}$ , if  $(u_{i1}, u_{i2})$  uses  $(v_1, v_2)$ , then  $(u_{j1}, u_{j2})$  can not use  $(v_1, v_2)$ . Lemma 2 now follows. ■

**Lemma 3:** *Let  $(v_1, v_2)$  be an edge of dimension  $s$  in  $H$ . Let  $(u_{i1}, u_{i2})$  be a single-edge in set S2 of  $T'_i$  and let  $(u_{j1}, u_{j2})$  be a single-edge in set S2 of  $T'_j$ , for  $i \neq j \neq s+1$  and  $0 \leq i, j \leq k-1$ . If  $(v_1, v_2)$  is used by  $(u_{i1}, u_{i2})$ , then it can not be used by  $(u_{j1}, u_{j2})$ .*

**Proof:** We know that  $(u_{i1}, u_{i2}) = (0_s 0 1^{k-\alpha-3} 0_i *^\alpha, 1_s 0 1^{k-\alpha-3} 0_i *^\alpha)$  and  $(u_{j1}, u_{j2}) = (0_s 0 1^{k-\beta-3} 0_j *^\beta, 1_s 0 1^{k-\beta-3} 0_j *^\beta)$ , for  $\alpha = s-i-1$  and  $\beta = s-j-1$ . The proof is similar to the one of Lemma 2 and is omitted. ■

We next show that given any dimension  $s$ ,  $0 \leq s \leq k-1$ , an edge of dimension  $s$  in  $H$  is used by at most two double-edges and one single-edge in the embedding of  $T'_0, T'_1, \dots, T'_{k-1}$ . Let  $(v_1, v_2)$  be an edge of dimension  $s$  in  $H$ . Lemma 1 described how trees  $T'_s$  and  $T'_{s+1}$  use  $(v_1, v_2)$  and it characterized  $(v_1, v_2)$  to belong to either  $H_s^1, H_s^2, H_s^3$ , or  $H_s^4$ . The next four lemmas show that there exists at most one  $T'_i$ ,  $i \neq s, s+1$ , such that an edge  $e_i$  of  $T'_i$  uses  $(v_1, v_2)$ . In each of the lemmas, we need only consider the usage of  $(v_1, v_2)$  by an edge  $e_i$  that belongs to either R1, R2, S1, or S2 of  $T'_i$ . If  $e_i$  were to belong to L1 (resp. L2), then  $i = s$  (resp.  $i = s+1$ ).

**Lemma 4:** *If  $(v_1, v_2) \in H_s^1$ , then there exists at most one  $T'_i$ ,  $i \neq s, s+1$ , such that  $T'_i$  uses  $(v_1, v_2)$  for a single-edge either from R2 or from S2.*

**Proof:** Since  $(v_1, v_2) \in H_s^1$ , we have  $(v_1, v_2) = (0_s 0 *^{k-3} 0, 1_s 0 *^{k-3} 0)$  and edge  $(v_1, v_2)$  is used by a double-edge of  $T'_s$  and by a double-edge of  $T'_{s+1}$ . Let  $(u_{i1}, u_{i2})$  be an edge of  $T'_i$  that uses edge  $(v_1, v_2)$ . We distinguish 3 cases depending on whether  $(u_{i1}, u_{i2}) \in R1, R2, S1$ , or S2.

**Case 1:** Edge  $(u_{i1}, u_{i2}) \in R1$  or S1.

When  $(u_{i1}, u_{i2}) \in R1$ , we have  $(u_{i1}, u_{i2}) = (1_{i+2} 1^{k-3} 0_i 1, 0_{i+2} 1^{k-3} 0_i 1)$ . Since  $T'_{s-2}$  is the only tree which uses an edge of dimension  $s$  in set R1, we have  $i = s-2$  and thus  $(u_{i1}, u_{i2}) = (u_{(s-2)1}, u_{(s-2)2}) = (1_s 1^{k-3} 0_{s-2} 1, 0_s 1^{k-3} 0_{s-2} 1)$ . If  $(u_{i1}, u_{i2}) \in S1$ , then  $(u_{i1}, u_{i2}) = (1_s 1^{k-\alpha-3} 0_i *^\alpha 0, 0_s 1^{k-\alpha-3} 0_i *^\alpha 0)$  for  $\alpha = s - i - 2$  and  $1 \leq \alpha \leq k-4$ . In both of the cases we have  $b_{s+1} = 1$ , while  $b_{s+1} = 0$  in  $(v_1, v_2)$ . Thus  $(u_{i1}, u_{i2})$  can not be in R1 or S1.

**Case 2:** Edge  $(u_{i1}, u_{i2}) \in R2$ .

$T'_{s-1}$  is the only tree which uses an edge of dimension  $s$  in set R2 and hence we have  $i = s-1$  and  $(u_{i1}, u_{i2}) = (1_s 0 1^{k-3} 0_{s-1}, 0_s 0 1^{k-3} 0_{s-1})$ . Thus,

there exists exactly one edge in  $H_s^1$  which is also used by  $T'_{s-1}$ .

**Case 3:** Edge  $(u_{i1}, u_{i2}) \in S2$ .

In this case  $(u_{i1}, u_{i2}) = (0_s 01^{k-\alpha-3} 0_i *^\alpha, 1_s 01^{k-\alpha-3} 0_i *^\alpha)$ , for  $\alpha = s - i - 1$  and  $1 \leq \alpha \leq k - 3$ . Hence, there exist edges  $(v_1, v_2) \in H_s^1$  that are also used by edges in S2 of some  $T'_i$ . From Lemma 3, it follows that when  $(u_{i1}, u_{i2}) \in S2$  uses  $(v_1, v_2)$ , then no other single-edge  $(u_{j1}, u_{j2}) \in S2$  uses  $(v_1, v_2)$ . Hence, there exists at most one  $T'_i$  such that its single-edge  $(u_{i1}, u_{i2})$  belonging to S2 uses  $(v_1, v_2)$ .

It remains to be shown that Cases 2 and 3 can not happen simultaneously over an edge  $(v_1, v_2)$ . This is easily seen by observing that when  $(v_1, v_2)$  is used by a single-edge in set S2 of  $T'_i$ , then  $i$  is not equal to  $s - 1$ , i.e.,  $T'_i \neq T'_{s-1}$ . It now follows that, in addition to the double-edges of  $T'_s$  and  $T'_{s+1}$ , at most one single-edge of  $T'_i$  uses  $(v_1, v_2)$  and hence Lemma 4 follows. ■

**Lemma 5:** *If  $(v_1, v_2) \in H_s^2$ , then there exists at most one  $T'_i$ ,  $i \neq s, s+1$ , such that  $T'_i$  uses  $(v_1, v_2)$  for a double-edge from set S1.*

**Proof:** Since  $(v_1, v_2)$  belongs to  $H_s^2$ , we know that  $(v_1, v_2) = (0_s 1 *^{k-3} 0, 1_s 1 *^{k-3} 0)$ . Furthermore, edge  $(v_1, v_2)$  is used by a double-edge of  $T'_s$  and by a single-edge of  $T'_{s+1}$ . As in Lemma 4, we check whether  $(v_1, v_2)$  is also used by an edge of  $T'_i$  from sets R1, R2, S1, and S2. Let  $(u_{i1}, u_{i2})$  be an edge of  $T'_i$ ,  $i \neq s, s+1$ . Recall that if  $(u_{i1}, u_{i2})$  belongs to R1, then  $(u_{i1}, u_{i2}) = (u_{(s-2)1}, u_{(s-2)2}) = (1_s 1^{k-3} 0_{s-2} 1, 0_s 1^{k-3} 0_{s-2} 1)$ . If  $(u_{i1}, u_{i2})$  belongs to R2, then  $(u_{i1}, u_{i2}) = (u_{(s-1)1}, u_{(s-1)2}) = (1_s 01^{k-3} 0_{s-1}, 0_s 01^{k-3} 0_{s-1})$ , and if  $(u_{i1}, u_{i2})$  belongs to S2, then  $(u_{i1}, u_{i2}) = (0_s 01^{k-\alpha-3} 0_i *^\alpha, 1_s 01^{k-\alpha-3} 0_i *^\alpha)$ , for  $\alpha = s - i - 1$  and  $1 \leq \alpha \leq k - 3$ . Hence, we have  $b_{s-1} = 1$  in  $(u_{i1}, u_{i2}) \in R1$

and  $b_{s+1} = 0$  in  $(u_{i1}, u_{i2}) \in R2$  or  $S2$ . But  $b_{s-1} = 0$  and  $b_{s+1} = 1$  in  $(v_1, v_2)$  and  $(u_{i1}, u_{i2})$  can thus not be in  $R1$ ,  $R2$ , or  $S2$ .

If  $(u_{i1}, u_{i2})$  belongs to  $S1$ , then  $(u_{i1}, u_{i2}) = (1_s 1^{k-\alpha-3} 0_i *^\alpha 0, 0_s 1^{k-\alpha-3} 0_i *^\alpha 0)$  for  $\alpha = s-i-2$  and  $1 \leq \alpha \leq k-4$ . Hence, there exist edges  $(v_1, v_2) \in H_s^2$  that are also used by edges in  $S1$  of some  $T'_i$ . We know from Lemma 2 that double-edges  $(u_{i1}, u_{i2})$  and  $(u_{j1}, u_{j2})$  belonging to  $S1$ , for  $i \neq j$ , can not both use the same edge  $(v_1, v_2)$ . Hence, in addition to a double-edge of  $T'_s$  and a single-edge of  $T'_{s+1}$ , edge  $(v_1, v_2)$  is used by at most one double-edge  $(u_{i1}, u_{i2}) \in S1$ . ■

**Lemma 6:** *There exists only one edge  $(v_1, v_2) \in H_s^3$  that is used by an edge not in  $T'_{s+1}$  or  $T'_s$ . This edge is used by the double-edge in  $R1$  of  $T'_{s-2}$ .*

**Proof:** We have  $(v_1, v_2) = (0_s 1 *^{k-3} 1, 1_s 1 *^{k-3} 1)$  and every edge  $(v_1, v_2)$  is used by a single edge of  $T'_{s+1}$ . From Lemma 4, recall the label of edge  $(u_{i1}, u_{i2})$  when it belongs to  $R1$ ,  $R2$ ,  $S1$ , and  $S2$ . It is easy to see that exactly one of the edges (namely, edge  $(0_s 1^{k-3} 0 1, 1_s 1^{k-3} 0 1)$ ) from set  $H_s^3$  is used by edge  $(u_{i1}, u_{i2})$  of  $T'_i$  belonging to  $R1$  and thus  $i = s-2$ . The argument to show that no edge of  $H_s^3$  is used by any edge of  $T'_i$  belonging to  $R2$ ,  $S1$ , and  $S2$  is similar to the arguments in Lemmas 4 and 5, and it is omitted. ■

**Lemma 7:** *If  $(v_1, v_2) \in H_s^4$ , then there exists at most one  $T'_i$ ,  $0 \leq i \leq k-1$ , that uses  $(v_1, v_2)$  for an edge from set  $S2$ .*

**Proof:** We have  $(v_1, v_2) = (0_s 0 *^{k-3} 1, 1_s 0 *^{k-3} 1)$  and no edge of  $T'_s$  and  $T'_{s+1}$  uses  $(v_1, v_2)$ . The proof is similar to that of Lemma 4 and is omitted. ■

From the previous lemmas it follows that given an edge  $(v_1, v_2)$  of dimen-



sion  $s$  in  $H$ ,  $0 \leq s \leq k-1$ , at most two double-edges and one single-edge of trees  $T'_0, T'_1, \dots, T'_{k-1}$  use  $(v_1, v_2)$  and hence the embedding achieves a load of 5. We thus have the following theorem.

**Theorem 8:** *Let  $T_i$  be a complete binary tree consisting of  $n-1$  PEs, for  $0 \leq i \leq r-1$ ,  $r \leq \log n$ . Then,  $r$  trees  $T_0, T_1, \dots, T_{r-1}$  can be embedded into a  $\log n$ -dimensional hypercube  $H$  consisting of  $n$  nodes so that the dilation is 2, load is 5, and every node in  $H$  is assigned at most  $r$  PEs with at most 1 PE from a tree  $T_i$ .*

We conclude this section by showing that our embedding of  $T_0, T_1, \dots, T_{k-1}$  is also an embedding of  $k$  leap trees. An  $(n-1)$ -PE leap tree  $P$  is an  $n$ -PE complete binary tree to which the following leap-edges are added. Processor  $j$  on level  $\alpha$  is connected to processor  $j+2^{\alpha-1}$  on level  $\alpha$ , for  $0 \leq j \leq 2^{\alpha-1}-1$  and  $1 \leq \alpha \leq k-1$ . See Figure 2 for an example of a leap tree when  $k=4$ . Let  $P_0, P_1, \dots, P_{k-1}$  be  $k$   $(n-1)$ -PE leap trees. Then the PEs of every  $P_i$  are embedded as the PEs in the  $T_i$ 's. That the leap-edges of  $P_i$  have a dilation of 1 is shown as follows. We know that the PEs on level  $\alpha$  of  $P_i$  are assigned to nodes  $0_i *^\alpha 01^{k-\alpha-2}$  in  $H$ ,  $1 \leq \alpha \leq k-2$ , and the leaf PEs of  $P_i$  are assigned to nodes  $1_i *^{k-1}$ . Furthermore, the PEs of level  $\alpha$  in the left subtree of  $P_i$  are assigned to nodes  $0_i 1 *^{\alpha-1} 01^{k-\alpha-2}$  and the PEs of level  $\alpha$  in the right subtree of  $P_i$  are assigned to nodes  $0_i 0 *^{\alpha-1} 01^{k-\alpha-2}$ . Hence, the leap-edges  $(j, j+2^{\alpha-1})$  on level  $\alpha$  of  $P_i$  are assigned to edges  $(0_i 1 *^{\alpha-1} 01^{k-\alpha-2}, 0_i 0 *^{\alpha-1} 01^{k-\alpha-2})$  of dimension  $i+1$  in  $H$ ,  $0 \leq j \leq 2^{\alpha-1}-1$  and  $1 \leq \alpha \leq k-2$ . Similarly, the leap-edges between the leaf PEs of  $P_i$  are assigned to edges  $(1_i 1 *^{k-2}, 1_i 0 *^{k-2})$  of dimension  $i+1$  in  $H$ .

Every edge of  $H$  is used by at most 5 non leap-edges of the  $P_i$ 's. Given any dimension  $s$ ,  $0 \leq s \leq k-1$ , the leap-edges that use edges of dimension  $s$  are the ones in the leap tree  $P_{s-1}$ . Leap tree  $P_{s-1}$  has  $2^{k-1} - 1$  leap-edges and every such edge uses one edge of dimension  $s$  and no two leap-edges of  $P_{s-1}$  use the same edge. It follows that every edge of dimension  $s$  in  $H$  is used by at most 6 edges of  $P_0, P_1, \dots, P_{k-1}$ . Hence, we have the following result.

**Theorem 9:** *Let  $P_i$  be a leap tree consisting of  $n-1$  PEs, for  $0 \leq i \leq r-1$ ,  $r \leq \log n$ . Then,  $r$  leap trees  $P_0, P_1, \dots, P_{r-1}$  can be embedded into a  $\log n$ -dimensional hypercube  $H$  consisting of  $n$  nodes so that the dilation is 2, load is 6, and every node in  $H$  is assigned at most  $r$  PEs with at most 1 PE from a leap tree  $P_i$ .*

### 3 Embedding Linear Arrays and Meshes

#### 3.1 Embedding $k$ Linear Arrays

In this section we show how to embed  $k$   $n$ -PE linear arrays  $L_0, L_1, \dots, L_{k-1}$  into  $H$  with a dilation of 1 and a load of 2. The load is optimal since the linear arrays contain a total of  $k(2^k - 1)$  edges while  $H$  contains  $k2^{k-1}$  edges. Our embedding satisfies the requirement that every node of  $H$  is assigned precisely 1 PE of  $L_i$ ,  $0 \leq i \leq k-1$ . The technique used to embed the  $k$  linear arrays resembles the one used for the binary trees in the previous section. From an embedding of a single linear array we obtain the embedding of the  $k$  linear arrays that distributes the load over all the edges of  $H$  evenly

by assigning to every  $L_i$  a mark-bit and by using the dimensions of  $H$  in a cyclic order. Any Gray Code gives an embedding of a linear array into  $H$  with a dilation and load of 1. We next define the Gray Codes used by our embedding.

Let  $GC_k$  be the  $2^k$  elements of the Gray Code on  $k$  bits obtained by the following recursive definition. Let  $GC_1 = \{0, 1\}$  and  $GC_{k-1} = \{g_0, g_1, \dots, g_{2^{k-1}-1}\}$ . Then the  $2^k$  elements of  $GC_k$  are:

$$g_0\mathbf{1}, g_1\mathbf{1}, \dots, g_{2^{k-1}-1}\mathbf{1}, g_{2^{k-1}-1}\mathbf{0}, g_{2^{k-1}-2}\mathbf{0}, \dots, g_0\mathbf{0}.$$

Let  $GS_k$  be the Gray Sequence determining which bits change between two adjacent elements in  $GC_k$ . Then,  $GS_k = GS_{k-1}, (k-1), GS_{k-1}$  with  $GS_1 = 0$ . Let  $GC_k(j)$  and  $GS_k(j)$  denote the  $j^{\text{th}}$  element of  $GC_k$  and  $GS_k$ , respectively.

We now describe how to embed  $L_0, L_1, \dots, L_{k-1}$  into  $H$ . Let  $GC_k^i(j)$  denote the bit string which is obtained by shifting the bit string  $GC_k(j)$  right by  $i$  positions with wrap-around; i.e., if  $GC_k(j) = b_0b_1 \dots b_{k-1}$ , then  $GC_k^i(j) = b_{k-i}b_{k-i+1} \dots b_{k-1}b_0b_1 \dots b_{k-i-1}$ . We embed  $L_i$  into  $H$  by assigning PE  $j$  of  $L_i$  to node  $GC_k^i(j)$  of  $H$ , for  $0 \leq j \leq 2^k - 1$ . Thus PE 0 of  $L_i$  is assigned to node  $1^i01^{k-i-1}$  and the edge  $(j, j+1)$  of  $L_i$  uses an edge of dimension  $(GS_k(j) + i) \bmod k$ . In Figure 3 we show the embeddings of  $L_0$  and  $L_2$  into  $H$  when  $k = 4$ . Obviously, the embedding of  $L_0, L_1, \dots, L_{k-1}$  into  $H$  achieves a dilation of 1 and every node of  $H$  is assigned precisely 1 PE of  $L_i$ , for  $0 \leq i \leq k-1$ . We next show that the embedding achieves a load of 2.

In order to show that every edge in  $H$  is used by at most two edges of  $L_0, L_1, \dots, L_{k-1}$ , we consider how the  $L_i$ 's use edges of dimension  $s$  in

$H$ ,  $0 \leq s \leq k - 1$ . Linear array  $L_s$  uses the Gray Sequence<sup>3</sup>  $GS_k \oplus s$  and in it dimension  $s$  occurs  $2^{k-1}$  times. Hence, every edge of dimension  $s$  in  $H$  is used by an edge of  $L_s$ . For  $1 \leq i \leq k - 2$ , linear array  $L_{s+i}$  uses edges of dimension  $s$  for  $2^{i-1}$  of its edges. They are of the form  $(0_s *^{i-1} 0_{s+i} 1^{k-i-2} 0, 1_s *^{i-1} 0_{s+i} 1^{k-i-2} 0)$ , where the labels of the nodes are again written as  $b_s b_{s+1} \dots b_{k-1} b_0 \dots b_{s-1}$  instead of  $b_0 b_1 \dots b_{k-1}$  and subscript  $s$  in the label indicates the bit position  $s$ . Linear array  $L_{s-1}$  uses edges of dimension  $s$  for  $2^{k-2}$  edges having form  $(0_s *^{k-2} 1, 1_s *^{k-2} 1)$ . There are  $2^{k-2}$  edges of dimension  $s$  in  $H$  that are used by both an edge of  $L_s$  and an edge of  $L_{s-1}$ . Let  $S$  be the set of edges not used by  $L_{s-1}$ ; i.e.,  $S = \{(0_s *^{k-2} 0, 1_s *^{k-2} 0)\}$ . It is easy to see that with the exception of one edge (namely the edge  $(0_s 1^{k-2} 0, 1_s 1^{k-2} 0)$ ), every edge in  $S$  is used by exactly one edge of some linear array  $L_{s+i}$  and by an edge of  $L_s$ . Hence, our embedding achieves a load of 2 and we have the following theorem.

**Theorem 10:** *Let  $L_0, L_1, \dots, L_{r-1}$  be  $r$  linear arrays each having length  $n$ ,  $r \leq \log n$ . Then  $L_0, L_1, \dots, L_{r-1}$  can be embedded into a  $\log n$ -dimensional hypercube  $H$  consisting of  $n$  nodes with a dilation of 1 and an optimal load of at most 2 so that every node of  $H$  is assigned only 1 PE of  $L_i$ , for  $0 \leq i \leq r - 1$ .*

### 3.2 Embedding $k$ Meshes

Assume now that  $n = 2^k$  for an even integer  $k \geq 2$ . Let  $M_0, M_1, \dots, M_{k-1}$  be  $k$  meshes, each of size  $\sqrt{n} \times \sqrt{n} = 2^{k/2} \times 2^{k/2}$ . It is well known that a mesh of size  $\sqrt{n} \times \sqrt{n}$  can be embedded into an  $n$ -node hypercube  $H$  with a

---

<sup>3</sup>  $\oplus$  in  $GS_k \oplus s$  denotes modulo  $k$  addition of  $s$  to every element of  $GS_k$ .

dilation of 1 and a load of 1 [16]. We show that  $k$  meshes can be embedded into  $H$  with a dilation of 1 and an optimal load of 4.

We describe how to embed meshes  $M_0, M_1, \dots, M_{k/2-1}$  into  $H$  with a dilation of 1 and a load of 2. The embedding of the  $k$  meshes is then obtained by simply using this embedding twice. Let  $(\alpha, \beta)$  be the PE in row  $\alpha$  and column  $\beta$  of  $M_i$ ,  $0 \leq \alpha, \beta \leq 2^{k/2} - 1$ . Let  $GC_{k/2}$  be the Gray Code of  $2^{k/2}$  elements on  $k/2$  bits and let  $GS_{k/2}$  be the Gray Sequence of  $2^{k/2} - 1$  elements on  $k/2$  bits, as defined in Section 3.1. One of the standard ways to embed a single mesh, say mesh  $M_0$ , into  $H$  is to use  $GC_{k/2}(\alpha)$  for row  $\alpha$  and use  $GC_{k/2}(\beta)$  for column  $\beta$  and to assign PE  $(\alpha, \beta)$  of  $M_0$  to node  $b_0b_1 \dots b_{k/2-1}b_{k/2} \dots b_{k-1} = GC_{k/2}(\alpha)GC_{k/2}(\beta)$ . That is, we use Gray Sequence  $GS_{k/2}$  for the edges in every column and use  $GS_{k/2} \oplus \frac{k}{2}$  for the edges in every row. Figure 4 shows the embedding of  $M_0$  into  $H$  when  $k = 4$ . Obviously, if we embed  $M_1, M_2, \dots, M_{k/2-1}$  into  $H$  in the way  $M_0$  is embedded, the load would be  $k/2$ . In order to achieve a load of 2 we use two mark-bits for every mesh and we use dimensions  $0, 1, \dots, \frac{k}{2} - 1$  for the columns and dimensions  $\frac{k}{2}, \frac{k}{2} + 1, \dots, k - 1$  for the rows in cyclic order.

Let  $GC_{k/2}^i(j)$  again denote the bit string which is obtained by shifting the bit string  $GC_{k/2}(j)$  right by  $i$  positions with wrap-around. By using  $GC_{k/2}^i(\alpha)$  for row  $\alpha$  of  $M_i$  and  $GC_{k/2}^i(\beta)$  for column  $\beta$  of  $M_i$ , we assign PE  $(\alpha, \beta)$  of  $M_i$  to node  $GC_{k/2}^i(\alpha)GC_{k/2}^i(\beta)$ ,  $0 \leq \alpha, \beta \leq 2^{k/2} - 1$ . In other words, we assign PE  $(0, 0)$  of  $M_i$  to node  $GC_{k/2}^i(0)GC_{k/2}^i(0) = 1^i 01^{k/2-1} 01^{k/2-i-1}$ . We then use Gray Sequence  $(GS_{k/2} \oplus i) \bmod \frac{k}{2}$  for the edges in every column of  $M_i$  and use Gray Sequence  $\frac{k}{2} \oplus ((GS_{k/2} \oplus i) \bmod \frac{k}{2})$  for the edges in every row of  $M_i$  to assign the remaining PEs of  $M_i$ .

Trivially, the embedding achieves a dilation of 1 and every node of  $H$  is assigned one PE of  $M_i$ ,  $0 \leq i \leq k/2-1$ . In order to show that the embedding achieves a load of 2, we first consider only the edges in the columns of the  $M_i$ 's. We show that the columns of the  $M_i$ 's are embedded into  $H$  with a load of 2 by partitioning the  $\frac{k}{2}2^{k/2}$  columns of  $M_0, M_1, \dots, M_{k/2-1}$  into  $2^{k/2}$  sets of  $\frac{k}{2}$  columns each. For  $0 \leq q \leq 2^{k/2} - 1$ , set  $S_q$  contains all the columns that have  $GC_{k/2}^0(q)$  as their rightmost  $k/2$  bits. Hence,  $S_q$  contains column  $q$  of  $M_0$ . Since  $GC_{k/2}^i$  is simply a permutation of the elements of  $GC_{k/2}^0$ ,  $S_q$  contains exactly one column from each mesh  $M_i$ . Set  $S_q$  can be viewed as containing  $\frac{k}{2}$  linear arrays of length  $2^{k/2}$  each. From the embedding of linear arrays it now follows that the  $\frac{k}{2}$  linear arrays of set  $S_q$  are embedded into  $H$  with a load of 2. Any other column  $\beta$  of  $M_i$ ,  $\beta \notin S_q$ , can not collide with the columns in set  $S_q$  since the last  $k/2$  bits in column  $\beta$  are different from the ones in columns of set  $S_q$ . Hence, all the  $\frac{k}{2}2^{k/2}$  columns of  $M_0, M_1, \dots, M_{k/2-1}$  are embedded into  $H$  with a load of 2. A similar argument shows that all the rows are also embedded with a load of 2. As stated earlier, edges in the rows can not collide with the edges in the columns because of the use of different dimensions for rows and columns, and hence the embedding of  $M_0, M_1, \dots, M_{k/2-1}$  into  $H$  achieves a load of 2.

By embedding mesh  $M_{k/2+i}$  in the same way as we embedded  $M_i$ ,  $0 \leq i \leq k/2-1$ , we have an embedding of  $M_0, M_1, \dots, M_{k-1}$  into  $H$  with a load of 4 and a dilation of 1. Note that any embedding of  $M_0, M_1, \dots, M_{k-1}$ , for  $k \geq 8$ , into  $H$  must achieve a load of at least 4 since the  $k$  meshes contain a total of  $2k(2^k - 2^{k/2})$  edges and  $H$  contains only  $k2^{k-1}$  edges. We now

can state the following result.

**Theorem 11:** *Let  $M_0, M_1, \dots, M_{r-1}$  be  $r$  meshes each of size  $\sqrt{n} \times \sqrt{n}$ ,  $r \leq \log n$ . Then  $M_0, M_1, \dots, M_{r-1}$  can be embedded into a  $\log n$ -dimensional hypercube  $H$  with a dilation of 1 and an optimal load of at most 4 so that every node of  $H$  is assigned precisely 1 PE of  $M_i$ , for  $0 \leq i \leq r - 1$ .*

## 4 Embedding Different Types of Networks

In Sections 2 and 3 we described embeddings with constant dilation and constant load when the  $r$  source networks  $G_0, G_1, \dots, G_{r-1}$  are of the same type. Assume now that we are given  $r$  source networks  $N_0, N_1, \dots, N_{r-1}$ , where the first  $r_1$  networks are complete binary trees, the next  $r_2$  are leap trees, the next  $r_3$  are linear arrays, and the final  $r_4$  are meshes,  $\sum_{i=1}^4 r_i \leq r$ . By embedding  $N_i$  in exactly the same way as we would embed the network  $N_i$  if all the  $r$  networks were of the same type as  $N_i$ , we achieve a dilation of at most 2 and a load of at most  $6 + 2 + 4 = 12$ . Note that the leap trees and the complete binary trees together give a load of 6.

By carefully analyzing and slightly modifying the given embeddings for the networks, one can reduce the load further. We only state one of the results in this direction. When we are given  $r_1$  complete binary trees and  $r - r_1$  linear arrays, the combined load can be kept at 5. This is achieved by embedding the  $T_i$ 's,  $0 \leq i \leq r_1 - 1$ , as described in Section 2 and changing the embedding for the linear arrays as follows. We assign the PE 0 of  $L_i$  to node  $0^{i-1}10^{k-i}$  and then assign PE  $j + 1$  of  $L_i$  by using dimension  $(GS_k(j) + i - 1) \bmod k$  for the edge  $(j, j + 1)$  of  $L_i$ , for  $0 \leq j \leq 2^k - 2$  and

$r_1 \leq i \leq r-1$ . A careful analysis now shows that the load in the embedding of  $T_0, \dots, T_{r_1-1}, L_{r_1}, \dots, L_{r-1}$  into  $H$  is 5 and the dilation is 2.

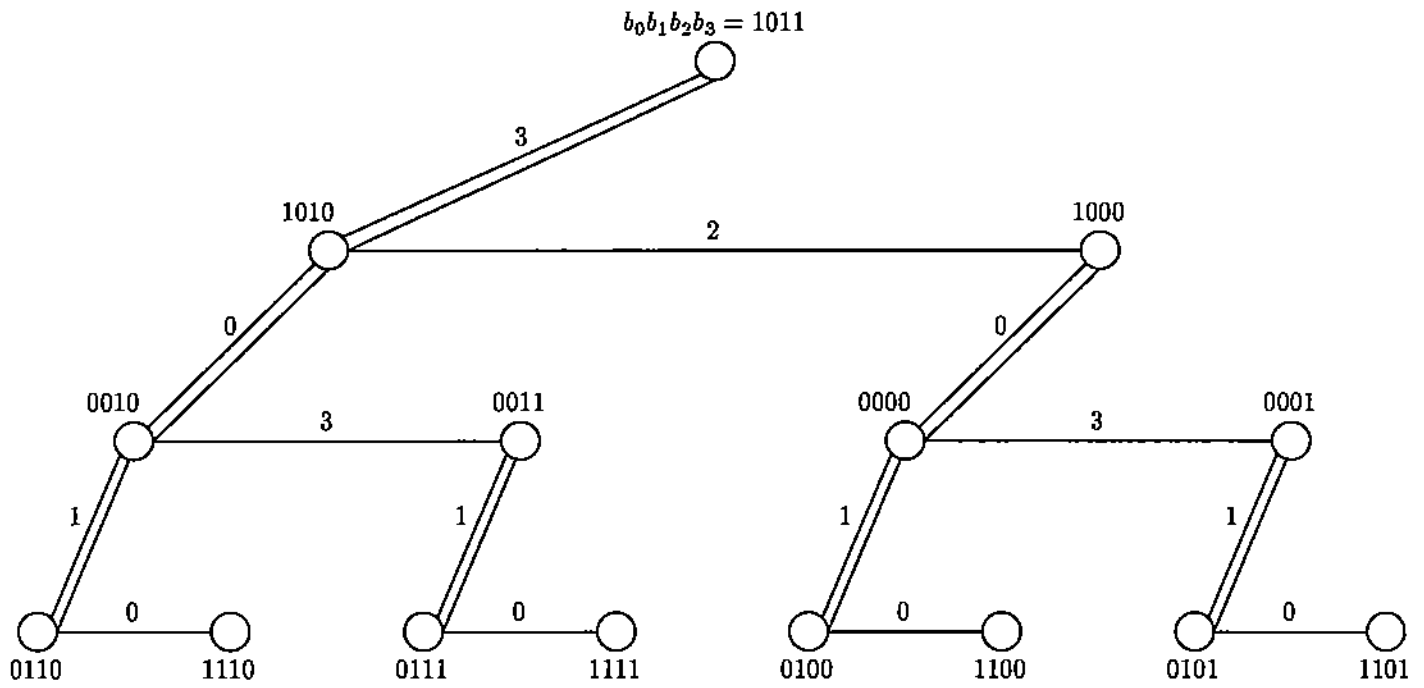
We conclude by stating that whereas we assumed throughout the paper that  $r$ , the number of source networks, is no greater than  $k$ , this is not a necessary constraint for our embeddings. If  $r > k$ , we simply partition the problem into  $\lceil \frac{r}{k} \rceil$  instances of the embedding problems described. The total load achieved now depends on  $\lceil \frac{r}{k} \rceil$ .

## References

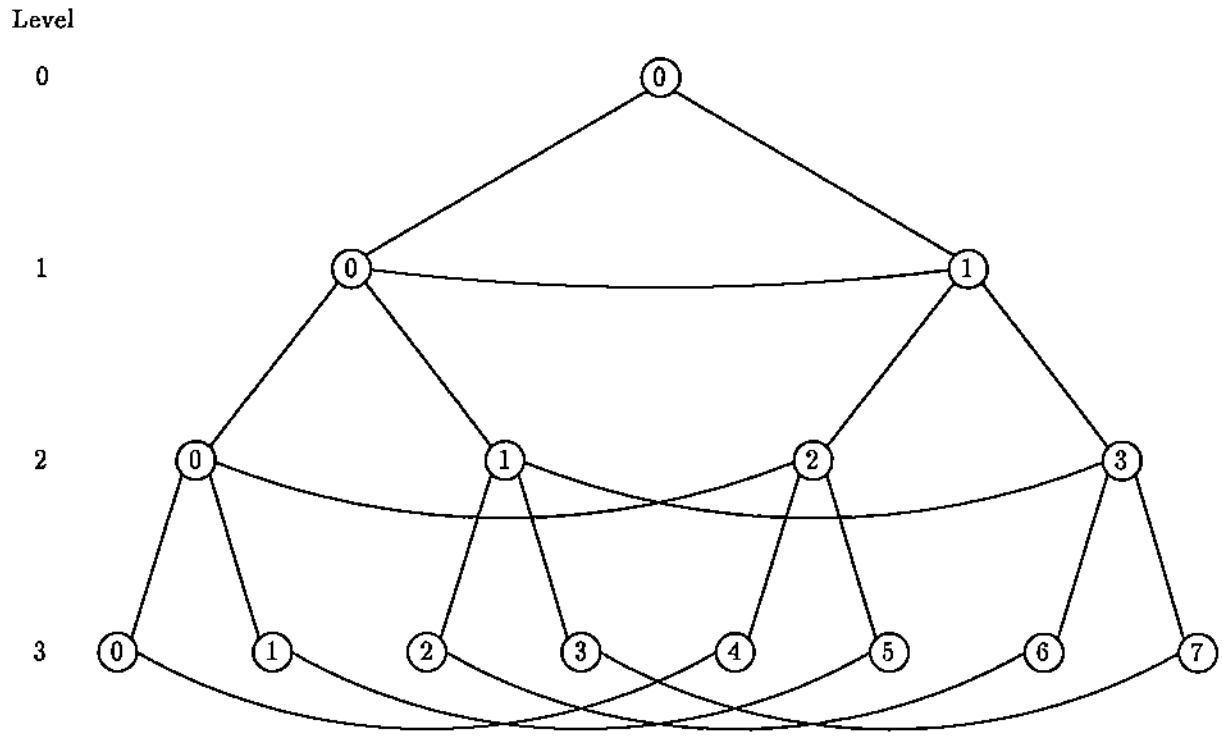
- [1] *NCUBE users Handbook*. NCUBE Co., 1987.
- [2] R. Aleliunas and A. Rosenberg. On embedding rectangular grids into square grids. *IEEE Trans. on Computers*, V-31:907-913, 1982.
- [3] M. J. Atallah. On multidimensional arrays of processors. *IEEE Transactions on Computers*, 37(10):1306-1309, October 1988.
- [4] S. Bettayeb, I. H. Sudborough, and Z. Miller. Embedding Grids into Hypercubes. In *the Proc. of 3<sup>rd</sup> Aegean Workshop on Computing*, 1988.
- [5] S. Bhatt, F. Chung, J. -W. Hong, F. T. Leighton, and A. L. Rosenberg. Optimal simulations by Butterfly networks. In *the Proceedings of 20<sup>th</sup> Symposium on Theory of Computing*, pages 192-204, May 1988.
- [6] S. Bhatt, F. Chung, F. T. Leighton, and A. Rosenberg. Optimal simulations of tree machines. In *the Proceedings of 27<sup>th</sup> Symposium on Foundation of Computer Science*, pages 274-282, 1986.
- [7] S. N. Bhatt and I. C. F. Ipsen. *How to Embed Trees in Hypercubes*. Research Report 443, Dept. of Computer Science, Yale University, December 1985.
- [8] M. Y. Chan and F. Y. L. Chin. On embedding rectangular grids in hypercubes. *IEEE Transactions on Computers*, 37(10):1285-1288, October 1988.



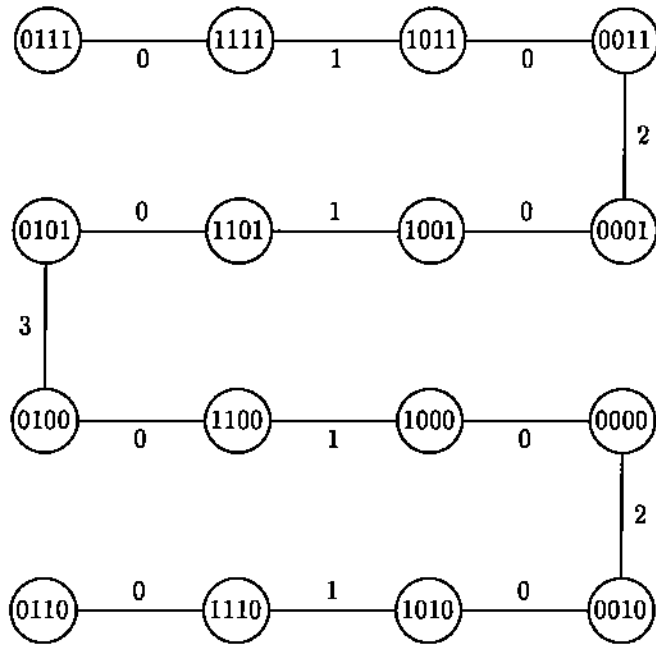
- [9] J. P. Fishburn and R. A. Finkel. Quotient networks. *IEEE Transactions on Computers*, C-31(4):288–295, April 1982.
- [10] A. K. Gupta and S. E. Hambrusch. *Embedding Complete Binary Trees into Butterfly Networks*. Technical Report CSD-TR-821, Purdue University, November 1988.
- [11] A. K. Gupta and S. E. Hambrusch. Embedding large tree machines into small ones. In *the Proceedings of 5<sup>th</sup> MIT Conference on Advanced Research in VLSI*, pages 179–199, March 1988.
- [12] W. D. Hillis. *The Connection Machine*. MIT Press, 1985.
- [13] J. -W. Hong, K. Mehlhorn, and A. Rosenberg. Cost trade-offs in graph embeddings, with applications. *JACM*, 709–728, 1983.
- [14] S. R. Kosaraju and M. Atallah. Optimal simulations between mesh connected array of processors. *Journal of the Association of Computing Machinery*, 35(3):635–650, July 1988.
- [15] A. L. Rosenberg. Preserving proximity in arrays. *SIAM Journal on Computing*, 443–460, 1979.
- [16] Y. Saad and M. H. Schultz. Topological properties of hypercubes. *IEEE Transactions on Computers*, 37(7):867–872, July 1988.



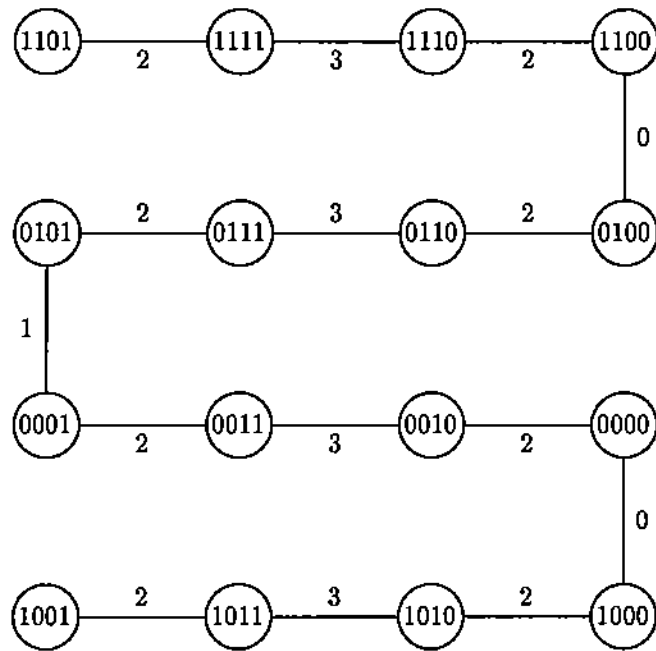
**Figure 1:** Embedding of  $T_1$  when  $k=4$ .



**Figure 2:** A Leap Tree when  $k=4$ .

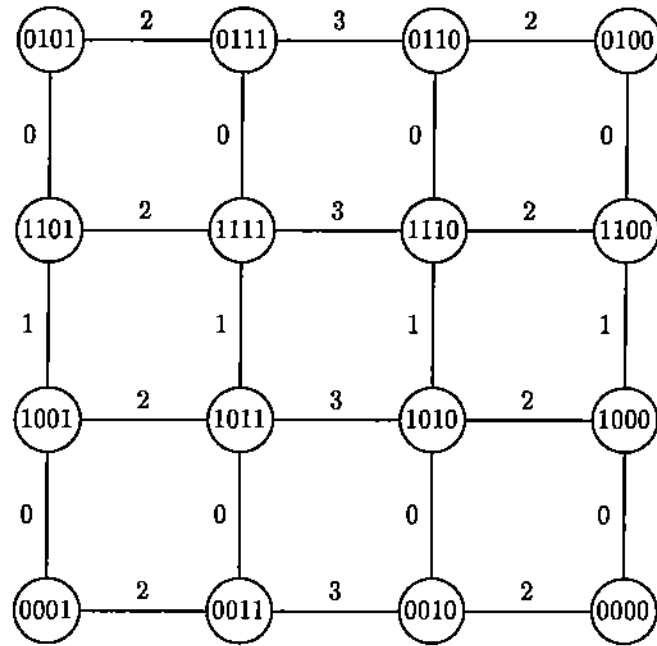


(a) Linear Array  $L_0$ .



(b) Linear Array  $L_2$ .

**Figure 3:** Embeddings of  $L_0$  and  $L_2$  when  $k=4$ . The numbers on the edges indicate the dimension of  $H$  used by that edge.



**Figure 4:** Embedding of  $M_0$  into  $C$  when  $k=4$ .