# Low-congestion Embedding of Multiple Graphs in a Hypercube*

Yu-Chee Tseng    Ten-Hwang Lai    Li-Fen Wu

Department of Computer and Information Science
The Ohio State University
Columbus, Ohio 43210
Tel: (614)292-5813, Fax: (614)292-9021
E-mail: tseng, lai, wu-l@cis.ohio-state.edu

December 3, 1992

**Abstract**

The purpose of this paper is to demonstrate the use of matrix as a representation of graph embedding in a hypercube. We denote the image of an embedding (which is a subgraph of the hypercube) as a matrix. Through such representation, we are able to simplify, unify, generalize, or improve existing results regarding multigraph embedding on a hypercube.

**Key Words:** hypercube, graph embedding, binary trees, meshes, regular binary-reflected trees, congestion, dilation.

*This research was supported by National Science Foundation under Grant CCR-9010589.

## 1 Introduction

The embedding of a guest graph $G$ into a host graph $H$ consists of two functions: an $f : V(G) \rightarrow V(H)$ mapping the vertex set of $G$ to that of $H$ in a one-to-one fashion, and a $g$ mapping each edge $\langle u, v\rangle \in E(G)$ to a path in $H$ that joins $f(u)$ to $f(v)$. The *dilation* of an edge $e \in E(G)$ is the length of $g(e)$, and the *dilation* of the embedding is the maximum dilation over all edges of $G$. The *congestion* of an edge $\langle x, y\rangle$ in $H$ is the number of edges $e \in E(G)$ such that $g(e)$ contains $\langle x, y\rangle$. The *congestion* of the embedding is the maximum congestion over all edges of $H$.

Embedding a guest graph $G$ into a host graph $H$ is an important problem in parallel and distributed computing, especially when the host graph is a hypercube. It's well known that meshes [2, 3, 12], pyramids [8], shuffle networks [13], and complete binary trees [14] all can be nicely embedded in hypercubes. On the other hand, hypercubes can be embedded in grids [7].

All these results involve only a single guest graph. Recently, multigraph embedding has received researchers' attentions. Johnsson and Ho [6] pointed out that $n$ binomial trees can be simultaneously embedded in an $n$-cube with congestion no more than two. Gupta and Hambrusch [5] obtained many interesting results: $n$ linear paths each of length $2^n - 1$ can be embedded with congestion 2 and dilation 1; $n$ complete level-$n$ binary trees can be embedded with congestion 5 and dilation 2; $n$ copies of $2^{n/2} \times 2^{n/2}$ meshes can be embedded with congestion 4 and dilation 1 (assuming an even $n$). All these results were obtained in an ad hoc fashion using different algorithms, in some cases very complex.

The main purpose of this paper is to demonstrate the use of matrix as a representation of graph embedding on a hypercube. In the literature, the embedding of a guest graph on a hypercube is typically described as a function or procedure, with the image of the guest graph under the embedding merely implicitly implied by the function or procedure. That often makes it very difficult to analyze the congestion of an embedding, especially in the case of multigraph embedding. In this paper, we propose a simple way to describe an embedding on a hypercube. Specifically, we represent the image of an embedding (which is a subgraph of the hypercube) as a matrix. Through such representation, we are able to identify a class of trees called *regular binary-reflected trees*, which includes linear paths, binomial trees, and many others. We show an interesting result that for any regular binary-reflected tree $T$, $n$ copies of $T$ can be simultaneously embedded in an $n$-cube with dilation 1 and congestion at most 2. This unifies and generalizes the above mentioned results regarding binomial trees and linear paths. Also, with matrix representation, we are able to embed $n$ complete level-$n$ binary trees with congestion 4, as opposed to congestion 5 of [5], and to generalize the embedding of multiple $2^{n/2} \times 2^{n/2}$ meshes [5] to general $k$-dimensional

$2^{d_1} \times \cdots \times 2^{d_k}$ meshes. Such generalization would be formidable without using our new representation method. Furthermore, our embedding is optimal for the whole class of regular binary-reflected trees and for meshes when $d_1 = \cdots = d_k$, in the sense that embedding more graphs is impossible without increasing the congestion.

The problem of multigraph embedding arises in a system that runs multiple jobs at a time. (Each job consists of a number of parallel tasks and is represented as a graph, with vertices representing tasks, and edges communication patterns.) One approach is to allocate a dedicated subsystem to each job. An alternative is to run all the jobs on the whole system, with each processor executing the tasks of different jobs in a time-sharing fashion. This paper assumes the latter approach, in which, by maintaining a sufficient number of ready-to-fire threads, the machine is able to better hide latency (e.g., message transmission, memory access and i/o delay), utilize computing power and balance load among nodes. Several multi-threaded architectures are in development, e.g., Alewife [1] and MIT's *T machine [10].

Section 2 introduces the concept of *edge matrix*, based on which the class of binary reflected trees is defined in Section 3. A subclass of the binary-reflected trees called regular binary-reflected trees is defined in Section 4, where its nice multi-embedding property is also shown. Section 5 considers the multi-embedding of complete binary trees. Section 6 demonstrates how to use edge matrices for multiple mesh embedding.

## 2 Edge Matrices

An $n$-dimensional hypercube is an undirected graph with $2^n$ nodes labeled from 0 to $2^n - 1$ (in binary representation); two nodes are joined by an edge iff their labels differ on exactly one bit. Thus, each node is represented by a binary string $b_0 b_1 \ldots b_{n-1}$, where $b_i \in \{0, 1\}$, and each $d$-dimensional subcube by a ternary string $x_0 x_1 \ldots x_{n-1}$, $x_i \in \{0, 1, *\}$, with exactly $d$ $*$'s, where $*$ means a "don't care" bit.

Two subcubes of the same dimension, say, $Q = x_0 \ldots x_{i-1} 0 x_{i+1} \ldots x_{n-1}$ and $Q' = x_0 \ldots x_{i-1} 1 x_{i+1} \ldots x_{n-1}$, are said to be *adjacent* along dimension $i$ if their string representations are different on exactly the $i$th bit. If $Q$ and $Q'$ are adjacent along dimension $i$, each node $b_0 \ldots b_{i-1} 0 b_{i+1} \ldots b_{n-1}$ in $Q$ is joined by an edge to a node $b_0 \ldots b_{i-1} 1 b_{i+1} \ldots b_{n-1}$ in $Q'$. The set of all such edges, denoted by $x_0 \ldots x_{i-1} \xi x_{i+1} \ldots x_{n-1}$, is referred to as the *edge set* between $Q$ and $Q'$, or simply an *edge set*. Note that the string representation of an edge set contains exactly one $\xi$, $d$ $*$'s, and $n - d - 1$ binary bits, where $d$ is the dimension of $Q$ and $Q'$. An edge set is said to be *full* if it is between two $(n-1)$-dimensional subcubes, or equivalently if its string representation contains only $*$'s besides the $\xi$. Throughout this paper, $n$ is the dimension of a given hypercube.

An *edge matrix* of an $n$-cube is a matrix with $m \geq 1$ rows and $n$ columns such that each row legitimately represents an edge set on the cube. If $X = (x_{i,j})$ is an edge matrix, let $E(X)$ denote the union of all of the edge sets represented by the rows of $X$, i.e.,

$$E(X) = \bigcup_{i\text{-th row}} x_{i,0}x_{i,1}\ldots x_{i,n-1},$$

where each set involved in the equation is regarded as a *multiset* and thus each element appears in $E(X)$ as many times as it appears in different edge sets on the right-hand side of the equation. The *multiplicity of an element* in a multiset is the number of times it appears in the set. The *multiplicity of a multiset* is the maximum multiplicity over all elements of the set.

**Example.** The following edge matrix specifies a Hamming path between two nodes 000111 and 110110.

$$\begin{pmatrix} \xi & 0 & 0 & 1 & 1 & 1 \\ 1 & \xi & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & \xi \end{pmatrix}$$

□

**Example.** As another example of the use of edge matrix, we make the following observation that will be used in subsequent sections:

$$**\ldots*\xi = E(\begin{pmatrix} b_0 & * & * & * & \ldots & * & * & \xi \\ \overline{b}_0 & b_1 & * & * & \ldots & * & * & \xi \\ \overline{b}_0 & \overline{b}_1 & b_2 & * & \ldots & * & * & \xi \\ \overline{b}_0 & \overline{b}_1 & \overline{b}_2 & b_3 & \ldots & * & * & \xi \\ & & & & \ldots & & & \\ \overline{b}_0 & \overline{b}_1 & \overline{b}_2 & \overline{b}_3 & \ldots & b_{n-3} & * & \xi \\ \overline{b}_0 & \overline{b}_1 & \overline{b}_2 & \overline{b}_3 & \ldots & \overline{b}_{n-3} & b_{n-2} & \xi \\ \overline{b}_0 & \overline{b}_1 & \overline{b}_2 & \overline{b}_3 & \ldots & \overline{b}_{n-3} & \overline{b}_{n-2} & \xi \end{pmatrix}), \qquad (1)$$

where $b_i \in \{0,1\}$, $0 \leq i \leq n-2$. This equation indicates how a full edge set can be decomposed into $n$ disjoint edge sets. Other decompositions can be obtained by permuting columns of the matrix on the right-hand side together with the bits on the left-hand side. □

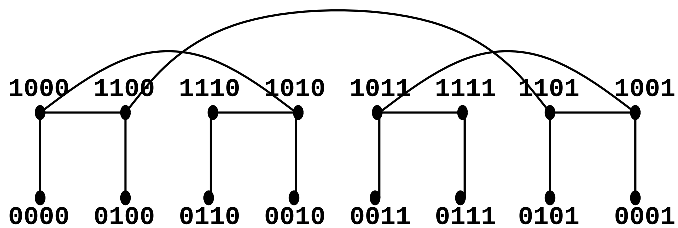
For an edge matrix $X$, define $G(X) = (V, E)$ to be the subgraph of the $n$-cube such that $V$ consists of all nodes of the $n$-cube and $E = E(X)$. Also, for $i \geq 0$, define $\rho^i(X)$ to be the matrix obtained from $X$ by cyclically shifting each column of $X$ to the right by $i$ positions. Note that $\rho^n(X) = \rho^0(X) = X$.

Denote by $X^{(i)}$ the $i$-th row of $X$. As a convention, the rows and columns of a matrix are numbered starting from 0.

Let $a \oplus b$ denote the "exclusive-or" of two bits $a \in \{*, \xi, 0, 1\}$ and $b \in \{0,1\}$ with the extension that $* \oplus b = *$ and $\xi \oplus b = \xi$ for $b = 0$ or 1. Define $(x_0, x_1, \ldots, x_{n-1}) \oplus$

$$X = \begin{pmatrix} \xi & * & * & * \\ 1 & \xi & * & * \\ 1 & 0 & \xi & * \\ 1 & 1 & 0 & \xi \end{pmatrix}$$

(a)




(b)

Figure 1: A $4 \times 4$ edge matrix defining a spanning tree in the 4-cube.

$(y_0, y_1, \ldots, y_{n-1}) = (x_0 \oplus y_0, \ldots, x_{n-1} \oplus y_{n-1})$. Given an $m \times n$ edge matrix $X = (x_{i,j})$ and a binary string $C$ of length $n$, define $X \oplus C$ to be also an $m \times n$ edge matrix $(X^{(0)} \oplus C, X^{(1)} \oplus C, \ldots, X^{(m-1)} \oplus C)^T$.

Two $m \times n$ edge matrices $X$ and $Y$ are said to be *isomorphic* if there exists a binary string $C$ of length $n$ such that $X$ can be obtained from $Y$ by permuting rows and/or columns of $Y$ (let the permuting result be $Y'$) followed by the operation $X = Y' \oplus C$. This definition is consistent with the common notion of *graph isomorphism*[1]. Indeed, it is not hard to see that if two edge matrices $X$ and $Y$ are isomorphic, then their corresponding graphs $G(X)$ and $G(Y)$ are isomorphic. We state this simple fact as a lemma for ease of future references.

**Lemma 1** *If $X$ and $Y$ are isomorphic edge matrices, then $G(X)$ and $G(Y)$ are isomorphic subgraphs of the hypercube.*

# 3 Binary-Reflected Trees

Many spanning trees on the $n$-cube can be described by $n \times n$ edge matrices. These include the well-known linear paths defined by the binary reflected Gray codes [4, 11], the binomial trees as define in [6], and many others. Fig. 1 shows an example edge matrix and the spanning tree defined by it. The matrix has non-$*$ bits only in the lower triangle. In general, we have the following.

[1]Two graphs $G$ and $H$ are *isomorphic* if there is a one-to-one correspondence $x \leftrightarrow x'$ between the nodes $x$ of $G$ and the nodes $x'$ of $H$ such that $\langle x, y \rangle$ is an edge in $G$ iff $\langle x', y' \rangle$ is an edge in $H$.

**Lemma 2** *Let $X = (x_{i,j})$ be an $n \times n$ edge matrix such that*

$$x_{i,j} = \begin{cases} \xi & \text{if } i = j \\ 0 \text{ or } 1 & \text{if } i > j \\ * & \text{if } i < j \end{cases} . \tag{2}$$

*Then $G(X)$ defines a spanning tree on the $n$-cube.*

**Proof**. Each edge set $X^{(i)}$ contains $2^{n-i-1}$ edges and so $E(X)$ contains $\sum_{i=0}^{n-1} |X^{(i)}| = \sum_{i=0}^{n-1} 2^{n-i-1} = 2^n - 1$ edges. Since $G(X)$ contains $2^n$ nodes, if we can show it to be connected then $G(X)$ is a spanning tree.

Claim: for any binary string $b_0 \ldots b_{n-1}$, all nodes in the subcube $* \ldots * b_i \ldots b_{n-1}$, where $0 \leq i \leq n$, are connected by $E(X)$. When $i = n$, the claim implies the connectivity of $G(X)$.

For $i = 0$, the claim is obviously true as $b_0 \ldots b_{n-1}$ contains only one node. Assume for the induction hypothesis that the claim is true for $i = k < n$ and consider the case $i = k + 1$. By the hypothesis, the nodes in subcube $A = * \ldots * 0 b_{k+1} \ldots b_{n-1}$ are connected by $E(X)$, and so are the nodes in $B = * \ldots * 1 b_{k+1} \ldots b_{n-1}$. The edge $x_{k,0} \ldots x_{k,k-1} \xi b_{k+1} \ldots b_{n-1}$ in $X^{(k)}$ joins a node in $A$ to a node in $B$. Thus, the nodes in the subcube $* \ldots * b_{k+1} \ldots b_{n-1}$ are connected by $E(X)$. The claim is proved. $\Box$

By Lemma 1, any edge matrix that can be transformed to a lower-triangular form defines a spanning tree on the hypercube. The following definition provides some insights into the structure of such a tree.

**Definition 1** A level-0 *binary-reflected* tree $T_0$ consists of a single node. A level-$k$ *binary-reflected* tree $T_k$, $k > 0$, is a tree consisting of two isomorphic $T_{k-1}$'s that are joined together by a single edge at any corresponding nodes (i.e., the endpoints of the edge are corresponding nodes of these two $T_{k-1}$'s under graph isomorphism).

Note that the binary-reflected trees are abstract trees, while the spanning trees $G(X)$ of the previous lemma are subgraphs of hypercubes with each node labeled by a binary string. However, the above definition does indicate how $G(X)$ is constructed. One can show by induction on $k = 0..n$ that the first $k$ rows of $X$ define $2^{n-k}$ connected components (each of which is a binary-reflected tree $T_k$ and all of which are isomorphic) in the cube and thus when $k = n$, $G(X)$ is a binary-reflected tree. Conversely, given any $T_n$, one can add a label to each node of its first subgraph $T_{n-1}$ with $x0$ and that of its second subgraph $T_{n-1}$ with $x1$, where $x$ is to be determined recursively and "consistently" in the sense that the corresponding nodes (under graph isomorphism) of the two $T_{n-1}$'s should have the same $x$; for the recursion to terminate, the single node of $T_0$ is labeled with an empty string. $T_n$ so labeled really uses a lower-triangular edge matrix. We thus have the following.

**Proposition 1** *The set of all level-$n$ binary-reflected trees is topologically equivalent to the set of all graphs $G(X)$ defined by some $n \times n$ edge matrix $X$ which is isomorphic to some lower-triangular edge matrix.*

All level-$n$ binary-reflected trees have the same number of nodes, $2^n$, but may have different heights. These trees contain the useful topologies of binomial trees (defined below) and linear paths, which respectively represent the shortest and the tallest trees in this class. Linear paths are common data structures; binomial trees have important applications in parallel divide-and-conquer algorithms [9]. In the following, we show how to find the corresponding edge matrices that define these topologies.

**Definition 2** An $n$-level binomial tree $B_n$ is a special $n$-level binary-reflected tree. $B_0$ consists of a single node, which is also its root. $B_i, i \geq 1$, is constructed out of two $B_{i-1}$'s with their roots joined by an edge and one of the two roots becoming the root of $B_i$.

**Lemma 3** *$G(X)$ is a binomial tree if the edge matrix $X$ is isomorphic to*

$$X_b = \begin{pmatrix} \xi & * & * & * & \ldots & * & * & * \\ b_0 & \xi & * & * & \ldots & * & * & * \\ b_0 & b_1 & \xi & * & \ldots & * & * & * \\ b_0 & b_1 & b_2 & \xi & \ldots & * & * & * \\ b_0 & b_1 & b_2 & b_3 & \ldots & * & * & * \\ & & & & \ldots & & & \\ b_0 & b_1 & b_2 & b_3 & \ldots & b_{n-3} & \xi & * \\ b_0 & b_1 & b_2 & b_3 & \ldots & b_{n-3} & b_{n-2} & \xi \end{pmatrix} \qquad (3)$$

*where $b_i \in \{0, 1\}$, $0 \leq i \leq n-2$.*

**Proof**. Without loss of generality, assume $X = X_b$. It is straightforward to prove by induction on $i$ that each edge in $b_0 \ldots b_{i-1}\xi * \ldots *$, say edge $b_0 \ldots b_{i-1}\xi x_{i+1} \ldots x_{n-1}$, joins two binomial trees $T_i$ (i.e., $* \ldots * 0 x_{i+1} \ldots x_{n-1}$ and $* \ldots * 1 x_{i+1} \ldots x_{n-1}$) at their roots and one of the roots, $b_0 \ldots b_{i-1} b_i x_{i+1} \ldots x_{n-1}$, becomes the root of $T_{i+1}$ (i.e. $* \ldots * \, x_{i+1} \ldots x_{n-1}$). This is consistent with the definition of binomial trees. □

**Lemma 4** *$G(X)$ is a Hamiltonian path if the edge matrix $X$ is isomorphic to*

$$X_h = \begin{pmatrix} \xi & * & * & * & \ldots & * & * & * \\ b_0 & \xi & * & * & \ldots & * & * & * \\ \overline{b}_0 & b_1 & \xi & * & \ldots & * & * & * \\ \overline{b}_0 & \overline{b}_1 & b_2 & \xi & \ldots & * & * & * \\ \overline{b}_0 & \overline{b}_1 & \overline{b}_2 & b_3 & \ldots & * & * & * \\ & & & & \ldots & & & \\ \overline{b}_0 & \overline{b}_1 & \overline{b}_2 & \overline{b}_3 & \ldots & b_{n-3} & \xi & * \\ \overline{b}_0 & \overline{b}_1 & \overline{b}_2 & \overline{b}_3 & \ldots & \overline{b}_{n-3} & b_{n-2} & \xi \end{pmatrix} \qquad (4)$$

*where $b_i \in \{0,1\}$, $0 \le i \le n-2$. (On each column, the bit immediately below $\xi$ is the complement of each of the bits below it.)*

**Proof**. Again, assume $X = X_h$. Since $G(X)$ is already known to be a spanning tree, we only have to show that each node in $G(X)$ has at most two edges incident upon it. If a node $x_0 x_1 \ldots x_{n-1}$ is incident on by an edge in $E(X)$ that is along dimension $i \ne 0$ of the hypercube, then $x_0 \ldots x_{i-1} = \overline{b}_0 \ldots \overline{b}_{i-2} b_{i-1}$; this indicates that along non-zero dimensions a node is incident on by at most one edge. Including edges in dimension 0, each node has a degree no more than two. $\square$

It is true that not every Hamiltonian path on the cube can be represented as an $n \times n$ matrix. Fortunately, the most important ones, that is, those which follow the Gary codes as defined below, do can be so represented.

**Definition 3** [4] Let $L = (l_0, l_1, \ldots, l_{n-1})$ be any permutation of $(0, 1, \ldots, n-1)$, $r_i$ be the partial rank of $l_i$ in $L$ (i.e., $r_i = |\{l_j : l_j < l_i, j < i\}|$), and $B = (b_0, \ldots, b_{n-1})$ be any sequence of 0 and/or 1. The Gray code $G_n$ defined by $L$ and $B$ is recursively defined as follows:

$$G_1 = (b_0, \overline{b}_0),$$
$$G_k = ((G_{k-1})^{b_k \backslash r_k}, (G^*_{k-1})^{\overline{b}_k \backslash r_k}), \qquad 1 \le k \le n-1,$$

where $G^*_{k-1}$ is the reversed sequence of $G_{k-1}$, and $(G_{k-1})^{b_k \backslash r_k}$ is the sequence obtained from $G_{k-1}$ by inserting a bit $b_k$ to each string of $G_{k-1}$ in the position that has $r_k$ bits in ahead of it.

**Lemma 5** *The Gray code $G_n$ defined by $L = (0, 1, \ldots, n-1)$ and $B = (b_0, \ldots, b_{n-1})$ can be represented by the following edge matrix:*

$$X_g = X_h \oplus (1, 1, \ldots, 1) = \begin{pmatrix} \xi & * & * & * & \ldots & * & * & * \\ \overline{b}_0 & \xi & * & * & \ldots & * & * & * \\ b_0 & \overline{b}_1 & \xi & * & \ldots & * & * & * \\ b_0 & b_1 & \overline{b}_2 & \xi & \ldots & * & * & * \\ b_0 & b_1 & b_2 & \overline{b}_3 & \ldots & * & * & * \\ & & & & \ldots & & & \\ b_0 & b_1 & b_2 & b_3 & \ldots & \overline{b}_{n-3} & \xi & * \\ b_0 & b_1 & b_2 & b_3 & \ldots & b_{n-3} & \overline{b}_{n-2} & \xi \end{pmatrix}. \qquad (5)$$

*The Gray codes defined with $L$ permuted can be represented by the edge matrix $X'_g$ that is column-permuted from matrix $X_g$ with the same permutation.*

# 4 Regular Binary-Reflected Trees

Among all binary-reflected trees, we are most interested in those which are *regular* in the following sense.

**Definition 4** An edge matrix $X$ is *regular* if it is isomorphic to some edge matrix $X_r$ of the following form:

$$X_r = \begin{pmatrix} \xi & * & * & * & \dots & * & * & * \\ b_0 & \xi & * & * & \dots & * & * & * \\ b_1 & b_0 & \xi & * & \dots & * & * & * \\ b_2 & b_1 & b_0 & \xi & \dots & * & * & * \\ b_3 & b_2 & b_1 & b_0 & \dots & * & * & * \\ & & & & \dots & & & \\ b_{n-3} & b_{n-4} & b_{n-5} & b_{n-6} & \dots & b_0 & \xi & * \\ b_{n-2} & b_{n-3} & b_{n-4} & b_{n-5} & \dots & b_1 & b_0 & \xi \end{pmatrix} \tag{6}$$

where $b_i \in \{0,1\}$, $0 \le i \le n-2$. A binary-reflected tree is *regular* if it is isomorphic to some $G(X)$ such that $X$ is regular.

In the following, we show that both binomial trees and linear paths are regular. We also prove that each of the Gray codes can be represented by a regular edge matrix.

**Proposition 2** *Binomial trees are regular.*

**Proof**. Let $b_0 = b_1 = \cdots = b_{n-1}$ in $X_r$ of Eq. (6). Such an $X_r$ conforms with one of the possible forms of $X_b$ in Eq. (3). $G(X_r)$ is a binomial tree, which proves this proposition. $\square$

**Proposition 3** *Linear paths of length* $2^n - 1$ *are regular.*

**Proof**. Let $b_0 = \overline{b}_1 = \overline{b}_2 = \cdots = \overline{b}_{n-1}$ in $X_r$ of Eq. (6). Such a graph $G(X_r)$ is Hamiltonian by observing $X_h$ in Eq. (4). $\square$

**Proposition 4** *Any Hamiltonian path defined by the Gray codes can be defined by a regular edge matrix.*

**Proof**. Without loss of generality, assume $L = (0, 1, \ldots, n-1)$. Recall Lemma 5 and observe that

$$X_g \oplus (b_0, b_1, \ldots, b_{n-1}) = \begin{pmatrix} \xi & * & * & * & \dots & * & * & * \\ 1 & \xi & * & * & \dots & * & * & * \\ 0 & 1 & \xi & * & \dots & * & * & * \\ 0 & 0 & 1 & \xi & \dots & * & * & * \\ 0 & 0 & 0 & 1 & \dots & * & * & * \\ & & & & \dots & & & \\ 0 & 0 & 0 & 0 & \dots & 1 & \xi & * \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & \xi \end{pmatrix}.$$

$X_g$ is regular as the right-hand side of the above equation is regular. $\square$

$$X_r = \begin{pmatrix} \xi & * & * & * \\ b_0 & \xi & * & * \\ b_1 & b_0 & \xi & * \\ b_2 & b_1 & b_0 & \xi \end{pmatrix}, X = \begin{pmatrix} \xi & * & * & * \\ b_0 & \xi & * & * \\ b_1 & \overline{b}_0 & \xi & * \\ b_2 & \overline{b}_1 & \overline{b}_0 & \xi \end{pmatrix}, \rho^1(X) = \begin{pmatrix} * & \xi & * & * \\ * & b_0 & \xi & * \\ * & b_1 & \overline{b}_0 & \xi \\ \xi & b_2 & \overline{b}_1 & \overline{b}_0 \end{pmatrix}$$

$$\rho^2(X) = \begin{pmatrix} * & * & \xi & * \\ * & * & b_0 & \xi \\ \xi & * & b_1 & \overline{b}_0 \\ \overline{b}_0 & \xi & b_2 & \overline{b}_1 \end{pmatrix}, \rho^3(X) = \begin{pmatrix} * & * & * & \xi \\ \xi & * & * & b_0 \\ \overline{b}_0 & \xi & * & b_1 \\ \overline{b}_1 & \overline{b}_0 & \xi & b_2 \end{pmatrix}$$

Figure 2: Four congestion-2 regular spanning trees in a 4-cube and their edge matrices.

It has long been known that as many as $n$ linear paths (each of length $2^n - 1$) or $n$ binomial trees can be embedded in an $n$-cube without causing a congestion more than two [5, 6]. In the following we show that not only these two kinds of graphs have this nice property but also does every regular binary-reflected tree, and, as a matter of fact, that is what makes regular binary-reflected trees so interesting.

**Theorem 1** *As many as $n$ copies of any regular binary-reflected tree can be simultaneously embedded in an $n$-cube with unit dilation and a congestion no more than two.*

**Proof**. In the following proof, we shall make use of the decomposition of a full edge set as in the second example of Section 2. Let $T$ be any regular binary-reflected tree. By definition, $T$ can be embedded with a regular edge matrix $X_r$ as in Eq. (6). Let $X = X_r \oplus (0, 1, \ldots, 1)$ and consider the $n$ mutually isomorphic edge matrices, $X, \rho(X), \rho^2(X), \ldots, \rho^{n-1}(X)$. Each of these defines a regular binary-reflected spanning tree isomorphic to $T$ (by Lemma 1). The congestion among these trees along dimension $j$ $(0 \leq j \leq n-1)$ is the multiplicity of the multiset $\cup_{i=0}^{n-1} (\rho^i(X))^{(j-i)} = \cup_{i=0}^{n-1} \rho^i(X^{(j-i)})$ (applying the modulo-$n$ operation to $j - i$ as appropriate). Observe that among these edge sets in the union, the $j$-th one (i.e., $\rho^j(X^{(j-j)}) = \rho^j(X^{(0)})$) is full and the rest $n-1$ ones are a "proper" decomposition of the full one (see Eq. (1)). The edge congestion along dimension $j$ is thus no more than 2. □

Fig. 2 shows the embedding procedure of the above proof on a 4-cube. These trees are all binomial trees when $b_0 = b_1 = b_2$, and all Hamiltonian paths when $b_0 = \overline{b}_1 = \overline{b}_2$.

Note that the congestion of two in the above result is inevitable even if only two copies of the tree are embedded — that is because the first copy already uses up one full edge set along some dimension while the second copy needs at least one edge along *each* dimension. Also note that $n$ is the maximum number of copies one may embed without raising the congestion to more than two — with congestion 2, the hypercube can accommodate at most $n2^n$ edges; with each spanning tree having $2^n - 1$ edges, that translates to $\lfloor n2^n/(2^n - 1) \rfloor = n$ copies at most when $n \geq 2$. Our embedding is optimal in this sense.

# 5 Embedding Multiple Complete Binary Trees

Denote by $T_n$ ($n > 0$)the $n$-level complete binary tree (with $2^n - 1$ nodes). A $T_n$ can be embedded in the $n$-cube with congestion 2 and dilation 2 [14], while $n$ copies of $T_n$ can be embedded with congestion 5 and dilation 2 [5]. Here we will improve the latter result by applying Theorem 1.

The basic idea is to embed a complete binary tree in a binomial tree with congestion 2 and dilation 2 and then apply the result for binomial trees in Theorem 1. That will result in an embedding of $n$ complete binary trees in the $n$-cube with congestion 4 and dilation 2.

Complete binary trees $T_i$ are embedded in binomial trees $B_i$ by the following algorithm. Note that $B_i$ has one more node than $T_i$.

**Algorithm** *Tree*$(T_i, B_i)$:

1. If $i = 1$, embed the only node of $T_1$ in the non-root node of $B_1$.
2. If $i > 1$, then
   (a) Recursively embed the two subtrees $T_{i-1}$ and $T'_{i-1}$ of $T_i$ into the two subtrees $B_{i-1}$ and $B'_{i-1}$ of $B_i$, respectively.
   (b) Suppose $x$ and $y$ are the roots of $B_{i-1}$ and $B'_{i-1}$, respectively, with $x$ being the root of $B_i$. Then, map the root of $T_i$ onto $y$; embed the edge between the roots of $T_i$ and $T'_{i-1}$ into the edge $\langle y, r(T'_{i-1}) \rangle$; and embed the edge between the roots of $T_i$ and $T_{i-1}$ into the path $y \to x \to r(T_{i-1})$, where $r(T)$ denotes the image of the root of a $T$ under the embedding. (See also Fig. 3.) □

The above procedure maintains an important property during the recursion: in *Tree*$(T_i, B_i)$, the root of $B_i$ is not used in the embedding and is adjacent to $r(T_i)$. This property makes the recursion possible. To see the congestion of the embedding, observe that the edge between $x$ and $y$ is used once in *Tree*$(T_i, B_i)$ and another time in *Tree*$(T_{i+1}, B_{i+1})$. Based on these discussions, it is straightforward to formally prove by induction that the above embedding incurs congestion 2 and dilation 2. We summarize the result of this section as follows.

**Theorem 2** *As many as $n$ copies of $T_n$ can be simultaneously embedded in the $n$-cube with congestion $\leq 4$ and dilation $\leq 2$.*

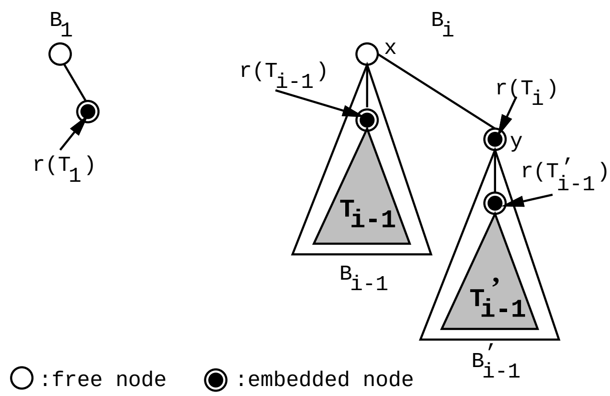



Figure 3: Embedding a complete binary tree on a binomial tree.

## 6 Embedding Multiple Meshes

This section demonstrates another application of edge matrices. Consider the problem of embedding in the $n$-cube multiple copies of a $k$-dimensional mesh of size $2^{d_1} \times 2^{d_2} \times \cdots \times 2^{d_k}$, where $d_1 + d_2 + \cdots + d_k = n$. A special case of the problem has been solved in [5]: when $k = 2$ and $d_1 = d_2$, $n$ meshes can be embedded in the $n$-cube with congestion 4 and dilation 1. That embedding is quite complex and seems formidable to generalize. Using edge matrices, we are able to develop a simpler and more general embedding.

A widely used technique for embedding a mesh in a hypercube is the "cross-product" of liner paths (e.g., see [12]). To embed a $2^{d_1} \times 2^{d_2} \times \cdots \times 2^{d_k}$ mesh in an $n$-cube, one first maps, for $1 \leq i \leq k$, a linear path of length $2^{d_i} - 1$ to a $d_i$-cube. Let $f_i$ be such a mapping. Then, each mesh node $(x_1, \ldots, x_k)$ is mapped to the hypercube node whose label is equal to the concatenation of $f_1(x_1), f_2(x_2), \ldots, f_k(x_k)$. Write this mapping as $f = f_1 \times f_2 \times \cdots \times f_k$. Note that in the above, any $f_i$, not necessarily those described in this paper, that embeds a linear path in a $d_i$-cube will work.

The embedding based on this mapping has unit congestion and unit dilation. For our multiple embedding purpose, we restrict $f_i$ as follows. Let $B_i'$ be a $d_i \times d_i$ matrix as defined in Eq. (6), with $b_0 = \overline{b}_1 = \overline{b}_2 = \cdots = \overline{b}_{d_i-2}$, and let $B_i = B_i' \oplus (0, 1, \ldots, 1)$. Similar to the proof of Proposition 3, $B_i$ defines a Hamiltonian path in a $d_i$-cube. Using this path as the image of the above mentioned embedding $f_i$, the cross-product embedding $f = f_1 \times \cdots \times f_k$ has the following $n \times n$ edge matrix:

$$X = \begin{pmatrix} B_1 & * & \ldots & * \\ * & B_2 & \ldots & * \\ \vdots & & \ddots & \vdots \\ * & * & \ldots & B_k \end{pmatrix} \tag{7}$$

where $*$ stands for a block of $*$'s. Fig. 4 illustrates an example matrix, with $b_0 =$

$$X = \begin{pmatrix} \xi & * & * & * & * & * & * \\ 1 & \xi & * & * & * & * & * \\ 0 & 0 & \xi & * & * & * & * \\ 0 & 1 & 0 & \xi & * & * & * \\ * & * & * & * & \xi & * & * \\ * & * & * & * & 1 & \xi & * \\ * & * & * & * & 0 & 0 & \xi \end{pmatrix}$$

Figure 4: The edge matrix of a $2^4 \times 2^3$ mesh.

$1, d_1 = 4$ and $d_2 = 3$.

Now let us return to the problem of embedding multiple $2^{d_1} \times \cdots \times 2^{d_k}$ meshes. Let $X$ be the matrix of Eq. (7). $G(X)$ is a $2^{d_1} \times \cdots \times 2^{d_k}$ mesh. Consider the $n$ isomorphic meshes

$$M_0, M_1, M_2, \ldots, M_{n-1} \tag{8}$$

where $M_i = G(\rho^i(X))$, $0 \leq i \leq n-1$. Since each $\rho^i(X)$ contains $k$ rows that are each full and since each $\rho^i(X)$ uses at least one edge along each dimension, the congestion of any two of these meshes is at least 2. The question now is how to select a largest subset $\Delta \subseteq \{M_0, \ldots, M_{n-1}\}$ with congestion no more than 2.

From Eq. (7), one observes that in each $\rho^i(X)$ exactly $k$ rows are full. Define $\mathcal{F}(M_i) = \{j|\ (\rho^i(X))^{(j)} \text{ is full}\}$; $\mathcal{F}(M_i)$ is thus the set of dimensions along which the edge sets of $M_i$ is full. The following lemma indicates a sufficient condition under which $\Delta$ has congestion no more than 2.

**Lemma 6** *Let $\Delta \subseteq \{M_0, M_1, \ldots, M_{n-1}\}$. If $\mathcal{F}(M_i) \cap \mathcal{F}(M_j) = \emptyset$ for any two distinct elements $M_i$ and $M_j$ in $\Delta$, then $\Delta$ has congestion no more than 2.*

**Proof.** Consider the congestion of $\Delta$ along dimension $r$ of the hypercube, where $0 \leq r \leq n-1$. By the given condition, there exists *at most one* $M_c$ in $\Delta$ which has a full edge set along dimension $r$.

*Case 1:* There is no $M_c$ in $\Delta$ that is full along dimension $r$. In this case, we claim that for any two distinct $M_i$ and $M_j \in \Delta$, the edge sets of $M_i$ and $M_j$ are disjoint along dimension $r$, from which it immediately follows that $\Delta$ has congestion 1 in dimension $r$. To verify the claim, let $M_i$ and $M_j$ be any two elements in $\Delta$. Observe from Eq. (7) that the edge set of $M_i$ (resp., $M_j$) along dimension $r$ has the string representation $A = \rho^i(* \ldots * b_s \overline{b}_{s-1} \ldots \overline{b}_0 \xi * \ldots *)$ for some $s > 0$ (resp., $B = \rho^j(* \ldots * b_t \overline{b}_{t-1} \ldots \overline{b}_0 \xi * \ldots *)$ for some $t > 0$) and $M_i$ is full along dimension $r-s$ (resp., $r-t$). Since no two elements in $\Delta$ are both full along the same dimension, it must be that $s \neq t$. $A$ and $B$, with their $\xi$-bit aligned, are readily seen to be complement of each other at the position of either bit $b_s$ of $A$ or $b_t$ of $B$. Thus, the two edge sets corresponding to $A$ and $B$ respectively are disjoint.

*Case 2:* There is an $M_c$ in $\Delta$ that is full along dimension $r$. In this case, $\Delta - \{M_c\}$ has congestion 1 along dimension $r$, as implied by *Case 1*, and so the congestion of $\Delta$ along the same dimension is 2. □

Let $d = \min\{d_1, d_2, \ldots, d_k\}$. It is not hard to see that $\Delta = \{M_0, M_1, \ldots, M_{d-1}\}$ satisfies the condition of Lemma 6 and hence has congestion no more than 2. (Fig. 5 shows such an example, using the edge matrix of Fig. 4). It is not known whether $|\Delta|$ is optimal except in the case of $d_1 = \cdots = d_k$.

**Theorem 3** *At least $m = \max\{2, \min\{d_1, d_2, \ldots, d_k\}\}$ copies of $2^{d_1} \times 2^{d_2} \times \cdots \times 2^{d_k}$ meshes can be simultaneously embedded into an $n$-cube with congestion no more than two. When $d_1 = d_2 = \cdots = d_k$, the number $m$ is optimal.*

**Proof**. The number of embedable meshes follows directly from the above comments. As for the optimality for the case of equilateral meshes, let $d = n/k$ be the common value of $d_1, \ldots, d_k$. If $d = 1$, the mesh becomes an $n$-cube and obviously no more than two $n$-cubes can be embedded in an $n$-cube with the limit of congestion 2. If $d \geq 2$, each mesh has $s = (2^d - 1) \cdot 2^{(k-1)d} \cdot k$ edges. With congestion 2, the $n$-cube can accommodate $s' = n2^n$ guest edges. It can be checked that $(d+1)s > s'$, and so no more than $d$ such meshes can be embedded in the $n$-cube without causing a congestion greater than two. □

For 2-dimensional meshes, we have a better result:

**Theorem 4** *The number of $2^{d_1} \times 2^{d_2}$ meshes that can be simultaneously embedded in an $n$-cube with congestion no more than 2 is at least $\lfloor \frac{1}{2} \cdot \frac{n}{\gcd(d_1, d_2)} \rfloor \cdot \gcd(d_1, d_2)$.*

**Proof**. For $0 \leq c, d < n$, it is a simple fact in number theory that the sequence $(c \bmod n, (c+d) \bmod n, (c+2d) \bmod n, (c+3d) \bmod n, \ldots)$ is periodic and contains exactly $\frac{n}{\gcd(d,n)}$ distinct elements. Simply write $(c, c+d, c+2d, \ldots)$ to denote the "cycle." The integers $0, 1, 2, \ldots, n-1$ can be partitioned into $\gcd(d, n)$ cycles, namely $(c, c+d, c+2d, \ldots)$ for $0 \leq c < \gcd(d, n)$. These cycles are pairwise disjoint.

With these backgrounds, consider the $\gcd(d_1, n)$ cycles $(M_c, M_{c+d_1}, M_{c+2d_1}, \ldots)$, $0 \leq c < \gcd(d_1, n)$, where the $M_i$'s are as defined in Eq. (8). Each mesh $M_i$ is full along dimensions $i$ and $i + d_1$ of the hypercube; i.e., $\mathcal{F}(M_i) = \{i, i + d_1\}$. Thus, two elements in a cycle have conflicts[2] iff they are consecutive in the cycle. Furthermore, these cycles are pairwise free of conflicts in the sense that meshes from different cycles do not conflict with each other. Each cycle has $\frac{n}{\gcd(d_1,n)}$ elements and is able to offer

[2]Two meshes $M_i$ and $M_j$ are said to have *conflicts* iff $\mathcal{F}(M_i) \cap \mathcal{F}(M_j) \neq \emptyset$.

$$X = \begin{pmatrix} \xi & * & * & * & * & * & * \\ 1 & \xi & * & * & * & * & * \\ 0 & 0 & \xi & * & * & * & * \\ 0 & 1 & 0 & \xi & * & * & * \\ * & * & * & * & \xi & * & * \\ * & * & * & * & 1 & \xi & * \\ * & * & * & * & 0 & 0 & \xi \end{pmatrix}$$

$$\mathcal{F}(M_0) = \{0, 4\}$$

$$\rho(X) = \begin{pmatrix} * & \xi & * & * & * & * & * \\ * & 1 & \xi & * & * & * & * \\ * & 0 & 0 & \xi & * & * & * \\ * & 0 & 1 & 0 & \xi & * & * \\ * & * & * & * & * & \xi & * \\ * & * & * & * & * & 1 & \xi \\ \xi & * & * & * & * & 0 & 0 \end{pmatrix}$$

$$\mathcal{F}(M_1) = \{1, 5\}$$

$$\rho^2(X) = \begin{pmatrix} * & * & \xi & * & * & * & * \\ * & * & 1 & \xi & * & * & * \\ * & * & 0 & 0 & \xi & * & * \\ * & * & 0 & 1 & 0 & \xi & * \\ * & * & * & * & * & * & \xi \\ \xi & * & * & * & * & * & 1 \\ 0 & \xi & * & * & * & * & 0 \end{pmatrix}$$

$$\mathcal{F}(M_2) = \{2, 6\}$$

Figure 5: The embedding of 3 $2^4 \times 2^3$ meshes.

$\lfloor \frac{1}{2} \cdot \frac{n}{\gcd(d_1,n)} \rfloor$ conflict-free meshes (e.g., taking every other element in the cycle). As $\gcd(d_1, n) = \gcd(d_1, d_2)$, the theorem is proved. □

It is not known whether the number $\lfloor \frac{1}{2} \cdot \frac{n}{\gcd(d_1,d_2)} \rfloor \cdot \gcd(d_1, d_2)$ established in the above theorem is optimal. We can show it to be at least nearly optimal though: with congestion 2, there are effectively $n2^n$ edges available in the hypercube, and as each mesh needs $2^{n+1} - 2^{d_1} - 2^{d_2}$ edges, the total number of meshes embedable is at most $\lfloor n2^n/(2^{n+1} - 2^{d_1} - 2^{d_2}) \rfloor = \lfloor n/(2 - 2^{-d_1} - 2^{-d_2}) \rfloor \approx \frac{n}{2} \approx \lfloor \frac{1}{2} \cdot \frac{n}{\gcd(d_1,d_2)} \rfloor \cdot \gcd(d_1, d_2)$.

Finally, note that although the results presented in this section are for meshes (which are the cross-product of linear paths) only, they can be applied to the cross-product of other regular binary-reflected trees identified in this paper as well. Simply observe that $B_i'$ used in the beginning of derivation can be a lower-triangular regular matrix of any kind.

# 7 Conclusions

We have demonstrated how to represent an embedding as an edge matrix and obtained results that simplify, unify, improve, or generalize existing results. The notion of edge matrix has proved to be useful especially when it comes to embedding multiple regular spanning trees. It's our belief that it can be applied to other embedding as well and is worth further study.

# References


[1] A. Agarwal, B.-H. Lim, D. Kranz and J. Kubiatowicz, "APRIL: A Processor Architecture for Multiprocessing," *Int'l Symp. on Computer Architecture*, 1990 pp. 104-114.

[2] M. Y. Chan and F. Y. L. Chin, "On Embedding Rectangular Grids in Hypercubes," *IEEE Trans. on Computers*, Vol. 37, No. 10, Oct. 1988, pp. 1285-1288.

[3] M. Y. Chan, "Embedding of $d$-Dimensional Grids into Optimal Hypercubes," *ACM Symposium of Parallel Architectures and Algorithms*, 1989, pp. 52-57.

[4] M.-S. Chen and K. G. Shin, "Processor Allocation in an $N$-Cube Multiprocessor Using Gray Codes," *IEEE Trans. on Computers*, Vol. C-36, No. 12, Dec. 1987, pp. 1396-1407.

[5] A. K. Gupta and S. E. Hambrusch, "Multiple Network Embeddings into Hypercubes," *Proc. of 5th Distributed Memory Computing Conf.*, Vol. II, 1990, pp. 1384-1392.

[6] S. L. Johnsson and C.-T. Ho, "Optimal Broadcasting and Personalized Communication in Hypercubes," *IEEE Trans. on Computers*, Vol. 38, No. 9, Sept. 1989, pp. 1249-68.

[7] T.H. Lai and A. Sprague, "Placement of the Processors of a Hypercube," *IEEE Trans. on Computers*, Vol. 40, No. 6, June 1991, pp. 714-722.

[8] T.H. Lai and W. White, "Mapping Pyramid Algorithms into Hypercubes," *J. of Para. and Distri. Comput.* **9**, pp. 42-54 (1990).

[9] V. M. Lo, *et al.*, "Mapping Divide-and-Conquer Algorithm to Parallel Architectures," *Int'l Conf. on Parallel Processing*, 1990, V. 3, pp. 128-135.

[10] R. S. Nikhil, G. M. Papadopoulos and Arvind, "*T: A Multithreaded Massively Parallel Architecture," *Int'l Symp. on Computer Architecture*, 1992, pp. 156-167.

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

[12] Y. Saad and M. H. Schultz, "Topological Properties of Hypercubes," *IEEE Trans. Comput.*, Vol. 37, No. 7, July 1988, pp. 867-872.

[13] R. Varadarajan, "Embedding Shuffle Networks in Hypercubes," *J. of Para. and Distri. Comput.* **11**, pp. 252-256 (1991).

[14] A. Y. Wu, "Embedding of Tree Networks into Hypercubes," *J. of Para. and Distri. Comput.* **2**, pp. 238-249 (1985).