On Embedding Various Networks Into The Hypercube Using Matrix Transformations

Mounir Hamdi[1]

Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
Email: hamdi@cs.ust.hk

Siang W. Song[2]

Department of Computer Science
University of Sao Paulo
C.P. 20570 - Sao Paulo
SP - 01452-990, Brazil

Abstract

Various researchers have shown that the binary n-cube (or hypercube) can embed any r-ary m-cubes, having the same number of nodes, with dilation 1. Their construction method is primarily based on the reflected Gray code. We present a different embedding method based on matrix transformations' schemes that achieves the same results. In addition, this method has a nice property that makes it suitable to be used in divide-and-conquer algorithms. Thus, it constitutes a useful tool for the design of parallel algorithms for the hypercube.

1 Introduction

The embedding of a guest graph G into a host graph H is an injection (one-to-one mapping) of the nodes in G to the nodes in H. the dilation cost of an embedding of G in H is the maximum distance in H between the images of any two neighboring nodes in G. This cost gives a measure of the proximity in H of the neighboring nodes in G under an embedding. Graph embedding results have many important applications in parallel processing. They provide the theoretical foundation for studying the problem of matching the communication structure of a task to the communication support of a parallel system and, also, for studying the problem of evaluating the relative performance of two interconnection networks. In particular, embedding various guest graphs into the the binary n-cube has received a lot of attention. It is well known that pyramids [4], shuffle networks [7], and complete binary trees [8] all can be nicely embedded in hypercubes.

[1]Supported in part by the Hong Kong Research Grant Council under the Grant RGC/HKUST 100/92E.

[2]Supported by FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) - Proc. No. 93/0603-1 and CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) - Proc. No. 306063/88-3 and PROTEM-CC/SP.

Of particular interest to us in this paper is the embedding of r-ary m-cube networks into hypercube. It is known [2, 3, 6] that the binary n-cube can embed any r-ary m-cube, with the same number of nodes, with dilation 1. These construction methods are based on the reflected Gray code. Using a different construction, we show a similar result, that any r-ary m-cube can be embedded in a binary n-cube with the same number of nodes with dilation 1. Our method is based on matrix transformation. Essentially we fold the hypercube recursively so that the final layout contains the r-ary m-cube. This layout has the following nice property. Once the binary n-cube has been laid out according to this method as an r-ary m-cube (with $2^n = r^m$), the sub-cubes formed by the nodes with the same first km bits, for $1 \leq k < \log r$, form smaller $(r/2^k)$-ary m-cubes. This property makes it suitable to be used in divide-and-conquer algorithms. As a result, it constitutes a useful tool for the design of parallel algorithms for the hypercube.

This paper is organized as follows. Section 2 gives a brief overview of our guest and host graphs. Section 3 presents the optimal embedding of r-ary m-cubes into the hypercube. Finally, in Section 4, we give some concluding remarks.

2 Overview of r-ary m-cube Networks

The binary n-cube is a special case of a family of r-ary m-cubes, cubes with m dimensions and r nodes on each dimension. Let N be the total number of nodes. Consider

$$N = r^m = 2^n \text{ or } m = \log_r N.$$

Denote by $(e_0, e_1, \ldots, e_{m-1})$, $0 \leq e_i < r$, the nodes of a r-ary m-cube. Two nodes e and e' are adjacent if

$$\begin{aligned} e &= (e_0, e_1, \ldots, e_i, \ldots, e_{m-1}) \\ e' &= (e_0, e_1, \ldots, e_i \pm 1 \bmod r, \ldots, e_{m-1}). \end{aligned}$$

For simplicity, we use also $i, j, \ldots$ instead of $e_0, e_1, \ldots$, for smaller dimensions. In the r-ary 2-cube, for example, node (i, j) is adjacent to nodes $(i, j \pm 1 \bmod r)$ and $(i \pm 1 \bmod r, j)$. (Figure 1 shows an 8-ary 2-cube.)

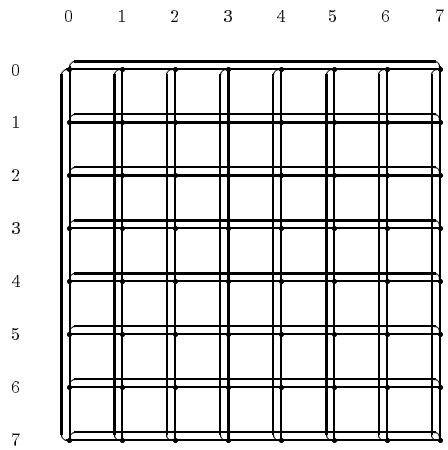

Figure 1: 8-ary 2-cube

3 Embedding by recursive folding

Let $N = r^m = 2^n$ and r a power of 2 ($r = 2^s$).

Consider as the guest graph G the r-ary m-cube and as the host graph H the binary n-cube.

Consider the nodes of H

$$0, 1, 2, \ldots, 2^{n-1}$$

and their binary representation

$$\underbrace{00\ldots00}_{n}, 00\ldots01, \ \ldots\ , 11\ldots10, 11\ldots11$$

Let us arrange these numbers in an m-dimensional matrix $\underbrace{r \times r \times \cdots \times r}_{m}$, denoted by $T(r, m)$. The elements of $T(r, m)$ are binary numbers of $m \log r$ bits each.

Consider the case of $m = 2$ and, for simplicity, write $T(r)$ instead of $T(r, 2)$. The general case will be handled later on.

3.1 Case $m = 2$

The nodes of the binary n-cube are laid out in a square $r \times r$ matrix denoted by

$$T(r) = (t_{ij}) = \begin{pmatrix} t_{00} & t_{01} & \ldots & t_{0,r-1} \\ t_{10} & t_{11} & \ldots & t_{1,r-1} \\ \vdots & \vdots & \ddots & \vdots \\ t_{r-1,0} & t_{r-1,1} & \ldots & t_{r-1,r-1} \end{pmatrix}$$

where $0 \le i, j < r$, $r = 2^{n/2}$ and each t_{ij} is a binary number of $n = m \log r$ bits.

To embed graph G in H, we use the following mapping g:

$$\text{node } (i, j) \text{ of } G \xrightarrow{g} \text{node } t_{ij} \text{ of } H.$$

Notation 1 *Denote by* $00T(r) = (t'_{ij})$ *where* $t'_{ij} = 00t_{ij}$, *that is,* 00 *followed by the bits of* t_{ij}. *In other words,* $00T(r)$ *is the matrix* $T(r)$ *with* 00 *in front of their elements.*

$$00T(r) = \begin{pmatrix} 00t_{00} & \ldots & 00t_{0,r-1} \\ \vdots & \ddots & \vdots \\ 00t_{r-1,0} & \ldots & 00t_{r-1,r-1} \end{pmatrix}$$

Analogously we have $01T(r)$, $10T(r)$ *and* $11T(r)$.

The bits that precede $T(r)$ *are said to be concatenated to* $T(r)$.

Notation 2 *Let*

$$T(r) = \begin{pmatrix} t_{00} & \ldots & t_{0,r-1} \\ t_{10} & \ldots & t_{1,r-1} \\ \vdots & \ddots & \vdots \\ t_{r-1,0} & \ldots & t_{r-1,r-1} \end{pmatrix} = \begin{pmatrix} l_0 \\ l_1 \\ \vdots \\ l_{r-1} \end{pmatrix}$$

where

$$l_i = \begin{pmatrix} t_{i0} & t_{i1} & \ldots & t_{i,r-1} \end{pmatrix} \ e \ c_j = \begin{pmatrix} t_{0j} \\ t_{1j} \\ \vdots \\ t_{r-1,j} \end{pmatrix}$$

We use the following notation:

$$T \downarrow\!\!\downarrow(r) = T(r)$$

$$T \downarrow\!\!\rightarrow(r) = \begin{pmatrix} c_{r-1} & \ldots & c_1 & c_0 \end{pmatrix}$$

$$T \rightarrow\!\!\downarrow(r) = \begin{pmatrix} l_{r-1} \\ \vdots \\ l_1 \\ l_0 \end{pmatrix}$$

$$T \rightarrow\!\!\rightarrow(r) = [T(r)]^T$$

where M^T *denotes the transpose of matrix* M.

The vertical arrow indicated the direction (from top to bottom or from bottom to top) index i of t_{ij} grows. The horizontal arrow indicates similarly the direction (from left to right or from right to left) index j of t_{ij} grows.

Definition 1 *We define $T(r)$ by means of the following recurrence.*

$$T(2) = \begin{pmatrix} 00 & 01 \\ 10 & 11 \end{pmatrix},$$

$$T(r) = \begin{pmatrix} 00T \overset{\downarrow}{\rightarrow}(r/2) & 01T \overset{\downarrow}{\leftarrow}(r/2) \\ 10T \overset{\uparrow}{\rightarrow}(r/2) & 11T \overset{\uparrow}{\leftarrow}(r/2) \end{pmatrix}, \quad r > 2.$$

Observe the following. Taking as *normal* index growth direction the direction $\overset{\downarrow}{\rightarrow}$(that is, index i from top to bottom and index j from left to right), we notice that each concatenated bit corresponds to a dimension, with 0 indicating the *normal* direction and 1 the inverse direction.

Theorem 1 *Let G be the r-ary 2-cube, with $r = 2^s$ and let H be the binary n-cube, with $n = 2\log r$. With the definition of $T(r)$ and using the mapping*

$$\textit{node } (i,j) \textit{ of } G \xrightarrow{g} \textit{node } t_{ij} \textit{ of } H,$$

we have an embedding of G in H of dilation 1, that is, adjacent nodes of G are mapped to adjacent nodes of H.

Proof:

By induction on r.
For $r = 2$,

$$T(2) = \begin{pmatrix} t_{00} & t_{01} \\ t_{10} & t_{11} \end{pmatrix} = \begin{pmatrix} 00 & 01 \\ 10 & 11 \end{pmatrix}.$$

The pairs of elements of $T(2)$, horizontal and vertical, are adjacent because they differ in exactly one bit.

Assume the theorem is valid for the case $r/2$. It is not difficult to show that the theorem holds for $T(r)$. By the induction hypothesis, we have in $T(r/2)$ or $T \overset{\downarrow}{\rightarrow}(r/2)$,

$$t_{ij} \text{ and } t_{i\pm 1 \ mod \ r,j} \text{ are adjacent and}$$

$$t_{ij} \text{ and } t_{i,j\pm 1 \ mod \ r} \text{ are adjacent.}$$

This is also true for $T \overset{\downarrow}{\leftarrow}(r/2), T \overset{\uparrow}{\rightarrow}(r/2)$ and $T \overset{\uparrow}{\leftarrow}(r/2)$, since the inversion of rows or columns does not alter the adjacency characteristics.

Thus, each one of $T \overset{\downarrow}{\rightarrow}(r/2)$, $T \overset{\downarrow}{\leftarrow}(r/2), T \overset{\uparrow}{\rightarrow}(r/2)$ and $T \overset{\uparrow}{\leftarrow}(r/2)$ forms a $(r/2)$-ary 2-cube.

Consider

$$T(r) = \begin{pmatrix} 00T \overset{\downarrow}{\rightarrow}(r/2) & 01T \overset{\downarrow}{\leftarrow}(r/2) \\ 10T \overset{\uparrow}{\rightarrow}(r/2) & 11T \overset{\uparrow}{\leftarrow}(r/2) \end{pmatrix}$$

$$= \begin{pmatrix} \text{first quadrant} & \text{second quadrant} \\ \text{third quadrant} & \text{fourth quadrant} \end{pmatrix}$$

Each one of the quadrants constitutes a sub-cube, whose nodes possess the same two initial bits. These sub-cubes form $(r/2)$-ary 2-cubes. Hence, it suffices to show that the four quadrants are connected through the boundaries (including the top, bottom, left and right boundaries).

Consider the last row of the first quadrant and the first row of the third quadrant. The corresponding elements of these rows differ only in the first bit, since, by definition, the last row of $T \overset{\downarrow}{\rightarrow}(r/2)$ is identical to the first row of $T \overset{\uparrow}{\rightarrow}(r/2)$. Thus they are adjacent. Similarly this holds for the other boundaries.

□

We thus have a method of arranging the nodes of a binary n-cube in the form of a r-ary 2-cube. Figure 2 is obtained directly from the definition of $T(r)$. It shows $T(8)$. At the center of each quadrant, we represent the two first bits of the elements belonging to the quadrant. Each quadrant, in turn, is divided into four quadrants, whose centers contain the two next bits of the elements belonging to them, and so on.

	0	1	2	3	4	5	6	7
0	00	01	01	00	00	01	01	00
	00		01		01		00	
1	10	11	11	10	10	11	11	10
		00				01		
2	10	11	11	10	10	11	11	10
	10		11		11		10	
3	00	01	01	00	00	01	01	00
4	00	01	01	00	00	01	01	00
	10		11		11		10	
5	10	11	11	10	10	11	11	10
		10				11		
6	10	11	11	10	10	11	11	10
	00		01		01		00	
7	00	01	01	00	00	01	01	00

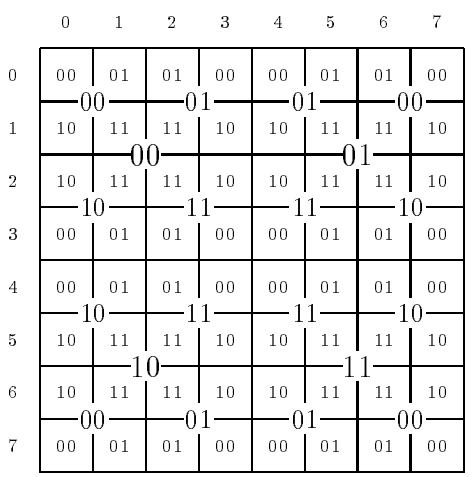

Figure 2: Layout of the nodes of the binary 6-cube by $T(8)$.

In Figure 3 we show the layout of the nodes of a binary 6-cube according to $T(8)$.

3.2 The general case: $m \geq 2$

What we have shown ($m = 2$) can be generalized to any case with $m \geq 2$. It is instructive to study the case $m = 3$, which can still be illustrated adequately.

Consider a binary n-cube and a r-ary 3-cube, $r = 2^s$ and with $n = 3\log r$. As before, we have the similar definition of $T(r,3)$, as shown in Figure 4. $T(r,3) = (t_{ijk})$ is a 3-dimensional $r \times r \times r$ matrix of binary numbers of $3\log r$ bits each.

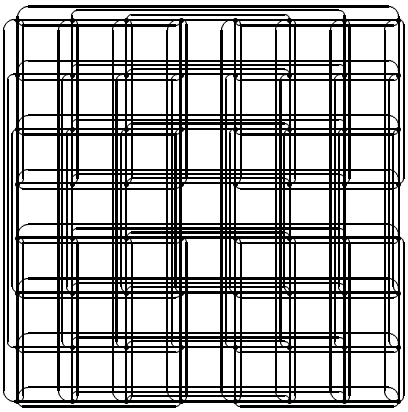

Figure 3: Layout of the binary 6-cube by $T(6)$

$T(r,3)$ is obtained from two copies of $T(r,2)$, one of which preceded by 0 and the other by 1. The additional dimension is represented by an additional arrow, denoted by $\nearrow$ or $\swarrow$, that represents the direction the new index k of t_{ijk} grows, according to the bit concatenated.

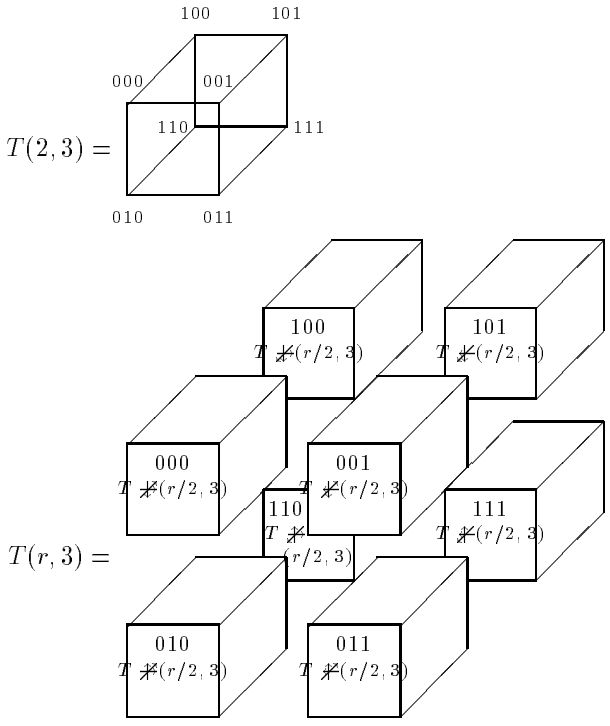

Figure 4: Definition of $T(r,m)$ for the case m=3

This construction can thus be extended to any m. Instead of using the more suggestive notation of arrows such as $\nearrow\!\!\!\!\searrow$etc., that do not make sense for higher dimensions, we can use a different notation such as, for example, $T\uparrow\uparrow\downarrow\uparrow\ldots$, in which each arrow corresponds to a dimension and represents a direction considered *normal* and another its inverse. We have the following theorem.

Theorem 2 *Let G be a r-ary m-cube, with $r = 2^s$ and let H be a binary n-cube, with $n = m \log r$. With $T(r,m)$ as constructed above and using the mapping*

$$\textit{node } (e_0, e_1, \ldots, e_{m-1}) \textit{ of } G \xrightarrow{g} \textit{node } t_{e_0,e_1,\ldots,e_{m-1}} \textit{ of } H,$$

we will have an embedding of G in H of dilation 1, with adjacent nodes of G mapped to adjacent nodes of H.

We have the following corollary.

Corollary 1 *If the nodes of a binary n-cube are laid out according to $T(r,m)$, then all the sub-cubes formed by the nodes with the same first km bits, for $1 \leq k < \log r$, are laid out as $(r/2^k)$-ary m-cubes.*

Based on the construction of $T(r)$, we can derive **Algorithm 1** that, given (i,j), obtains the corresponding node of the binary n-cube mapped to it by $T(r)$. For instance, given $(4,6)$, the algorithm gives 111001. Given $(3,6)$, it gives 011001.

Algorithm 1 Obtainment of g for the general case $m \geq 2$

$$\textit{Given } (e_0, e_1, \ldots, e_{m-1}),\ 0 \leq e_i < r,$$

$$\textit{obtain the node } d = d_0 d_1 \ldots d_{n-1},\ d_i = 0 \textit{ or } 1:$$

$$(e_0, e_1, \ldots, e_{m-1}) \xrightarrow{g} d_0 d_1 \ldots d_{n-1}.$$

```
begin
    for i := 0 to m − 1 do
        begin
            inf_i := 0;
            sup_i := r − 1;
            q_i := 0
        end;
    j := 0;
    k := 0;
    while j < log r  (ou n/m) do
        begin
        for i := 0 to m − 1 do
            begin
                if e_i < (inf_i + sup_i + 1)/2 then
                  begin
                  sup_i := (inf_i + sup_i + 1)/2 − 1;
                  t_i := 0
                  end
                else
                  begin
```

```
                    inf_i := (inf_i + sup_i + 1)/2;
                    t_i := 1
                    end;
                d_k := q_i ⊕ t_i;
                k := k + 1;
                q_i := t_i
            end;
        j := j + 1
        end
end
```

We also have **Algorithm 2**, that does the inverse of **Algorithm 1**: given a node of G obtain the corresponding node of H. Both these two algorithms have time complexities of $O(m \log r) = O(n) = O(\log N)$.

Algorithm 2 Obtainment of h for the general case $m \geq 2$

$$Given \quad d = d_0 d_1 \ldots d_{\bar{n}-1}, \; d_i = 0 \; or \; 1,$$

$$obtain \; the \; node \quad (e_0, e_1, \ldots, e_{\bar{m}-1}), \; 0 \leq e_i < r :$$

$$d_0 d_1 \ldots d_{\bar{n}-1} \xrightarrow{h} (e_0, e_1, \ldots, e_{m-1}).$$

```
begin
    for i := 0 to m − 1 do
        begin
            inf_i := 0;
            sup_i := r − 1;
            t_i := 0
        end;
    j := 0;
    k := 0;
    while j < log r  (ou n/m) do
        begin
        for i := 0 to m − 1 do
            begin
                t_i := t_i + d_k;
                if t_i = 0 then
                    sup_i := (inf_i + sup_i + 1)/2 − 1
                else
                    inf_i := (inf_i + sup_i + 1)/2;
                k := k + 1
            end;
        j := j + 1
        end;
    for i := 0 to m − 1 do
        e_i := inf_i (temos inf_i = sup_i)
end
```

4 Conclusions

The proposed embedding, based on matrix transformations, does not differ significantly from the inverted Gray code methods. It has nevertheless a new property, as stated in the Corollary. Once the nodes of a binary n-cube are laid out according to $T(r, m)$, then *all its sub-cubes*, formed by nodes with the *same* initial km bits, also form smaller m-cubes. This property does not hold in the inverted Gray code method. It may be interesting in designing parallel algorithms on the hypercube of the type divide-and-conquer, which is the objective of our work. It constitutes thus a useful tool for the design of parallel algorithms for the hypercube.

It is not surprising that the binary n-cube (graph H) contains the r-ary m-cube (graph G), when both graphs are of the same size: $r = 2^s$ and $n = m \log r$. It can be readily seen that graph H possesses more edges than graph G, when $s > 2$.

References

[1] Bertsekas, D.P. and Tsitsiklis, J.N., *Parallel and Distributed Computation - Numerical Methods*, Prentice-Hall, 1989.

[2] Chan, M. Y. and Chin, F. Y., "On embedding Rectangular Grids in Hypercubes," *IEEE Trans. Comput.*, p. 1285-1288, 1988.

[3] Chen, M. S. and Shin, K. G., "Processor Allocation in an N-cube Using Gray Codes," *IEEE Trans. Comput.*, p. 1396-1407, 1987.

[4] Lai, T. H. and White, W., "Mapping Pyramid Algorithms Into Hypercubes", *J. Parallel and Distributed Computing.*, p. 42-54, 1990.

[5] McBryan, O.A. and Van de Velde, E.F., "Hypercube Algorithms and Implementations", *SIAM J. Sci. Stat. Comput.*, Vol. 8, N. 2, p. s227-287, March 1987.

[6] Y. Saad and M. H. Schultz. Topological properties of hypercubes. *IEEE Transactions on Computers* 37 (7): 867 - 872, 1988.

[7] Varadarajan, R., "Embedding Shuffle Networks Into Hypercubes", *J. Parallel and Distributed Computing.*, p. 252-256, 1991.

[8] Wu, A. Y., "Embedding of Tree Networks Into Hypercubes", *J. Parallel and Distributed Computing.*, p. 238-249, 1985.