Discussiones Mathematicae
Graph Theory 28 (2008) 463–476

EMBEDDING COMPLETE TERNARY TREES INTO HYPERCUBES

S.A. Choudum and S. Lavanya

Department of Mathematics
Indian Institute of Technology Madras
Chennai 600 036, India

e-mail: sac@iitm.ac.in
e-mail: s.lavanya@yahoo.com

Abstract

We inductively describe an embedding of a complete ternary tree T_h of height h into a hypercube Q of dimension at most $\lceil (1.6)h \rceil + 1$ with load 1, dilation 2, node congestion 2 and edge congestion 2. This is an improvement over the known embedding of T_h into Q. And it is very close to a conjectured embedding of Havel [3] which states that there exists an embedding of T_h into its optimal hypercube with load 1 and dilation 2. The optimal hypercube has dimension $\lceil (\log_2 3)h \rceil$ $(= \lceil (1.585)h \rceil)$ or $\lceil (\log_2 3)h \rceil + 1$.

Keywords: complete ternary trees, hypercube, interconnection network, embedding, dilation, node congestion, edge congestion.

2000 Mathematics Subject Classification: 05C05, 05C90, 68E10.

1. Introduction

Graph embeddings constitute a central topic in the area of parallel and distributed computing; see [5, 6, 8]. They are natural mathematical models capturing the issues involved in the design of parallel algorithms. We assume that the reader is familiar with the terminology associated with graph embeddings. However, since the terminology is varied in the literature, we recall the most general definition of a graph embedding to avoid any confusion.

Let G and H be any two graphs and let $\wp(H)$ denote the set of all paths in H. An *embedding* of a guest graph $G(V, E)$ into a host graph $H(W, F)$

is a pair of functions (f, ρ) where $f : V \to W$ and $\rho : E \to \wp(H)$ such that ρ maps an edge uv of G to a path connecting $f(u)$ and $f(v)$ in H. The parameters load, dilation, node congestion, and edge congestion are associated with such an embedding (f, ρ) to measure its qualities.

The *load* of a node $v \in V(H)$ is the number of nodes of $V(G)$ that are mapped onto v by f; the *load* of (f, ρ) is the maximum load over all nodes of H. Note that, if f is an injective map, then the load is 1.

The *dilation* of an edge $e(uv)$ in G is the length of the path $\rho(e)$. The *dilation* of (f, ρ) is the maximum dilation over all edges of G. Note that, if there exists a load 1 and dilation 1 embedding of G into H, then G is isomorphic to a subgraph of H.

The *congestion of an edge* $e' \in H$ is the number of edges $e \in E(G)$ such that the path $\rho(e)$ contains e'. The *edge congestion* of (f, ρ) is the maximum congestion over all edges of H.

The *congestion of a node* $v \in H$ is the number of edges $e \in E(G)$ such that v is an internal vertex of the path $\rho(e)$. The *node congestion* of (f, ρ) is the maximum congestion over all nodes of H.

All embeddings discussed in this paper have load 1, that is f is an injection. Also, we map every edge $e(uv)$ onto a shortest $(f(u), f(v))$-path. Nevertheless, it is still important which paths we choose, since we are interested in obtaining an embedding with node congestion 2 and edge congestion 2.

There is a vast body of work on embedding of various kinds of trees into hypercubes. In particular, complete k-ary trees have received special attention as they represent algorithms that employ divide-and-conquer strategy.

A *complete k-ary tree* of height h, is a rooted tree in which each internal vertex has exactly k children and the distance from the root to each leaf is exactly h.

For $n \geq 1$, the *n-dimensional hypercube* (*or n-cube*), Q_n, is the graph whose vertex set is the set of binary strings $V(Q_n) := \{X := x_1x_2\dots x_n : x_i \in \{0,1\}, 1 \leq i \leq n\}$ and edge set $E(Q_n) := \{XY : X$ and Y differ in exactly one position $\}$. Alternatively, hypercubes are recursively defined through the cartesian product $(\times)$ of graphs as $Q_1 = K_2$, and for $n \geq 2$, $Q_n = Q_{n-1} \times K_2$. This definition permits a decomposition of Q_n into two copies of Q_{n-1}, say Q^0_{n-1} and Q^1_{n-1} as follows: $V(Q^0_{n-1}) = \{X \in V(Q_n) : X = 0x_2\dots x_n\}$ and $V(Q^1_{n-1}) = \{X \in V(Q_n) : X = 1x_2\dots x_n\}$. Any vertex $0x_2\dots x_n \in V(Q^0_{n-1})$ is adjacent to a unique vertex $1x_2\dots x_n \in V(Q^1_{n-1})$. Similarly, we can further decompose Q^0_{n-1} and Q^1_{n-1} and obtain four copies Q^{00}_{n-2}, Q^{01}_{n-2}, Q^{10}_{n-2} and Q^{11}_{n-2} of Q_{n-2}.

Given a tree T, let n be the smallest integer such that $2^n \geq |V(T)|$. Then Q_n is called the *optimal hypercube* of T, and Q_{n+1} is called the *next-to-optimal hypercube*.

Henceforth, T_h will denote a complete 3-ary tree (that is, a ternary tree) of height h. The root of T_h will be denoted by R_h. The three children of the root R_h, namely the left child, the middle child and the right child will be denoted by c_l, c_m and c_r, respectively. Since T_h has $(3^{h+1}-1)/2$ vertices, it follows that its optimal hypercube has dimension $\lceil(\log_2 3)h\rceil (\approx \lceil(1.585)h\rceil)$ or $\lceil(\log_2 3)h\rceil + 1$. The following conjecture is open since 1990.

Conjecture 1.1 (Havel [3])**.** Any complete ternary tree of height h can be embedded with load 1 and dilation 2 into its optimal hypercube.

The following result achieves the smallest dilation, node congestion, and edge congestion known so far for embedding T_h into a $\lceil(1.6)h\rceil + 1$-dimensional hypercube.

Theorem 1.2 (Gupta *et al.* [2])**.** *Any complete ternary tree of height h can be embedded with load 1, dilation 3 and edge congestion 3 into $Q_{d(h)}$, where $d(h) = \lceil(1.6)h\rceil + 1$.*

The related results on embedding of ternary trees into hypercubes can be found in many papers [1, 4, 7, 8].

In this paper, we prove the following improvement of the above theorem.

Theorem 1.3. *Any complete ternary tree of height h can be embedded with load 1, dilation 2, edge congestion 2 and node congestion 2 into $Q_{d(h)}$, where*

$$d(h) = \begin{cases} \lceil(1.6)h\rceil, & \text{if } h \equiv 2 \pmod 5 \text{ or } h \equiv 4 \pmod 5, \\ \lceil(1.6)h\rceil + 1, & \text{if } h \equiv 0 \pmod 5 \text{ or } h \equiv 1 \pmod 5 \text{ or } h \equiv 3 \pmod 5. \end{cases}$$

2. Embedding of Complete Ternary Trees

Let τ_h denote the tree obtained from T_h by adding a new vertex D and joining it to the root R_h of T_h. Here, we call the vertex D as the *deep root* of the tree τ_h. If a tree τ_h is embeddable into a hypercube Q with load 1, dilation 2, node congestion 2 and edge congestion 2, such that the root R_h and the deep root D are mapped onto adjacent nodes and the edge $(f(R_h), f(D)) \in Q$ has congestion 1, then we write $\tau_h \hookrightarrow Q$. We first prove the following result which implies our main result, Theorem 1.3, by induction on h, since T_h is a subgraph of τ_h. Its proof technique is a refinement of the technique employed in [2].

Theorem 2.1. *If* $\tau_h \hookrightarrow Q_d$, *then* $\tau_{h+5} \hookrightarrow Q_{d+8}$.

Proof. We prove the result by describing the following five embeddings.

(i) $\tau_{h+1} \hookrightarrow Q_{d+2}$,

(ii) $\tau_{h+2} \hookrightarrow Q_{d+4}$,

(iii) $\tau_{h+3} \hookrightarrow Q_{d+5}$,

(iv) $\tau_{h+4} \hookrightarrow Q_{d+7}$ and

(v) $\tau_{h+5} \hookrightarrow Q_{d+8}$.

(i) $\tau_{\mathbf{h+1}} \hookrightarrow \mathbf{Q_{d+2}}$: The embedding (f, ρ) of τ_{h+1} in Q_{d+2} is schematically shown in Figure 1.

To obtain this embedding, we first decompose Q_{d+2} into four copies of Q_d. For $i, j \in \{0, 1\}$, we have a copy of τ_h in Q_d^{ij} denoted by $ij\tau_h$; see Figure 1(b). So, we suitably combine these embeddings, as shown in Figure 1(c), to obtain an embedding of τ_{h+1}, with $10D$ as its deep root and $00D$ as its root. The embedding maps the three children c_l, c_m, c_r of the root R_{h+1} in T_{h+1} onto the vertices $01R_h$, $00R_h$ and $10R_h$ respectively, with the following properties.

(i) The edges $R_{h+1}c_l$ and $R_{h+1}c_r$ of T_{h+1} are mapped onto the paths $(00D, 01D, 01R_h)$ and $(00D, 00R_h, 10R_h)$ of Q_{d+2}, respectively. So, they have dilation 2. The edges $R_{h+1}c_m$ and $R_{h+1}D$ are mapped onto the edges $(00D, 00R_h)$ and $(00D, 10D)$, respectively. So, they have dilation 1. The edge joining the root R_{h+1} and the deep root D, (that is, the edge $(00D, 10D)$) has congestion 1. Note that the edge $(00D, 00R_h)$ has congestion 2, since it belongs to $\rho(R_{h+1}c_r)$ and $\rho(R_{h+1}c_m)$.

(ii) The edges of the three trees T_h rooted at c_l, c_m, c_r in T_{h+1}, retain their dilation attained in the embedding $\tau_h \hookrightarrow Q_d$. Every node and every edge of Q_{d+2} retains its congestion attained in the embedding $ij\tau_h \hookrightarrow Q_d^{ij}$, $i, j \in \{0, 1\}$.

Input:

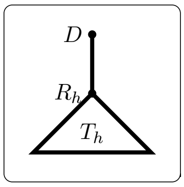

Figure 1(a). $\tau_h \hookrightarrow Q_d$. D and R_h are d-bit binary strings.

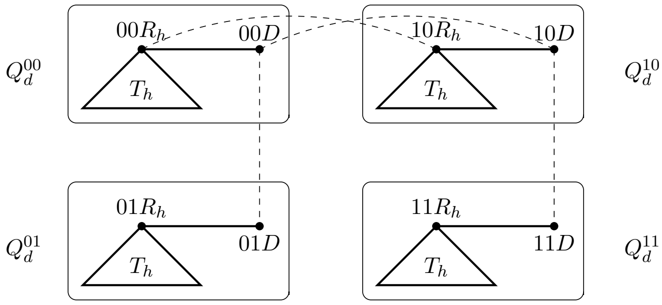

Figure 1(b). $ij\tau_h \hookrightarrow Q_d^{ij}, i, j \in \{0, 1\}$

Embedding:

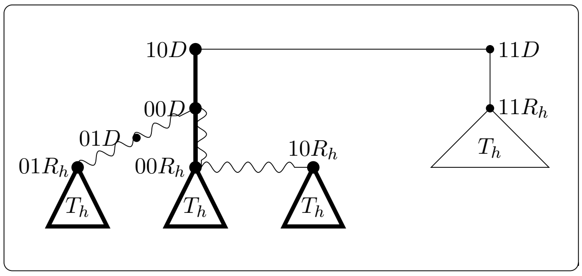

Figure 1(c). The edge $(00D, 00R_h)$ has congestion 2.

Output:

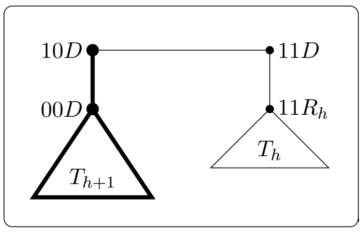

Figure 1(d). $\tau_{h+1} \hookrightarrow Q_{d+2}$. The τ_h shown on the right with light edges contains unutilized vertices.

Figure 1. The steps involved in an embedding of $\tau_{h+1} \hookrightarrow Q_{d+2}$, with the input $\tau_h \hookrightarrow Q_d$.

(iii) Figure 1(c) also shows a ternary tree τ_h (with $11D$ as the deep root and $11R_h$ as the root) embedded in Q_{d+2} whose vertices are not yet utilized. These vertices would be used in the subsequent embeddings. Here again, the edge $(11D, 11R_h)$ has congestion 1.

(iv) Every parameter namely, dilation, node congestion and edge congestion of this embedding is bounded by 2.

The output of this step, shown in Figure 1(d), is the input for embedding τ_{h+2} in Q_{d+4} in the next step.

(ii) $\tau_{\mathbf{h+2}} \hookrightarrow \mathbf{Q_{d+4}}$: The embedding (f, ρ) of τ_{h+2} in Q_{d+4} is described in Figure 2.

Similar to the previous step, we obtain an embedding of τ_{h+2} with $1010D$ as its deep root and $0010D$ as the root; see Figure 2(b). The embedding maps the three children c_l, c_m, c_r of the root R_{h+2} in T_{h+2} onto the vertices $0100D$, $0000D$, and $1000D$, respectively. The edges $R_{h+2}c_l$ and $R_{h+2}c_r$ of T_{h+2} have dilation 2. The edges $R_{h+2}c_m$ and $R_{h+2}D$ have dilation 1. The rest of the edges of T_{h+2} retain their dilation attained in the embedding $\tau_{h+1} \hookrightarrow Q_{d+2}$. Note that the node $0110D$ which appears twice (shown inside a circle) and the node $0111D$ which appears twice (shown inside a square) will receive node congestion 2 in the subsequent steps, as the nodes appear in the set of vertices which are yet to be utilized. The edge $\rho(R_{h+2}D) = (0010D, 1010D)$ receives congestion 1. Note also that the edge $(0010D, 0000D)$ has congestion 2, since it belongs to $\rho(R_{h+2}c_r)$ and $\rho(R_{h+2}c_m)$. The remaining nodes and edges of Q_{d+4} retain their congestion attained in the embedding $ij\tau_{h+1} \hookrightarrow Q^{ij}_{d+2}$, for $i, j \in \{0, 1\}$.

Figure 2(b) also shows two copies of τ_{h+1} (one with $1110D$ as the deep root and $1100D$ as the root τ_{h+1} and the second with $0111D$ as the deep root and $1111D$ as the root) and a copy of τ_h (with $0011D$ as the deep root and $0011R_h$ as the root) embedded in Q_{d+4} whose vertices are not yet utilized. In each case, the edge joining the root and the deep root of the tree are mapped onto adjacent nodes and receives congestion 1. Every parameter namely, dilation, node congestion and edge congestion of these embeddings is bounded by 2.

The output of this step, shown in Figure 2(c), is the input for the next step to embed τ_{h+3} in Q_{d+5}.

(iii) $\tau_{\mathbf{h+3}} \hookrightarrow \mathbf{Q_{d+5}}$: The embedding (f, ρ) of τ_{h+3} in Q_{d+5} is described in Figure 3.

Input: The output of the embedding $\tau_{h+1} \hookrightarrow Q_{d+2}$; see Figure 1(d).

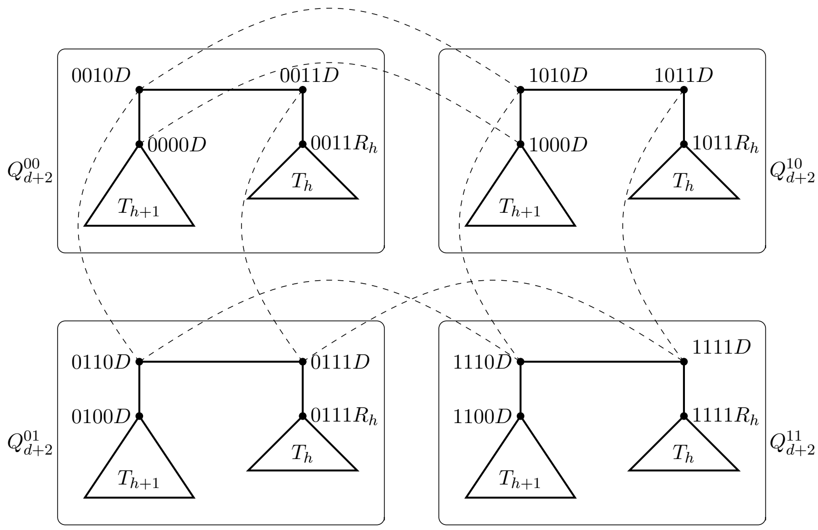

Figure 2(a). $ij\tau_{h+1} \hookrightarrow Q^{ij}_{d+2}, i, j \in \{0, 1\}$

Embedding:

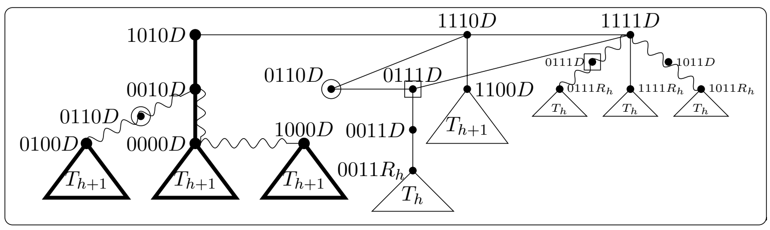

Figure 2(b). The edge $(0010D, 0000D)$ has congestion 2. The nodes $0110D$ and $0111D$ are the candidates to receive node congestion 2 in the subsequent steps.

Output:

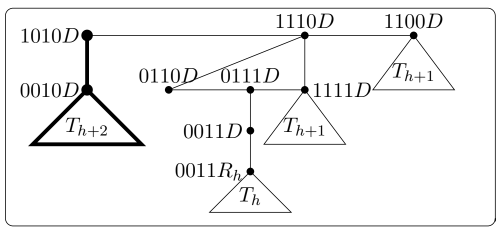

Figure 2(c). $\tau_{h+2} \hookrightarrow Q_{d+4}$. The subgraph shown on the right with light edges contains unutilized vertices.

Figure 2. The steps involved in an embedding of $\tau_{h+2} \hookrightarrow Q_{d+4}$, with the input $\tau_{h+1} \hookrightarrow Q_{d+2}$.

Input: The output of the embedding $\tau_{h+2} \hookrightarrow Q_{d+4}$; see Figure 2(c).

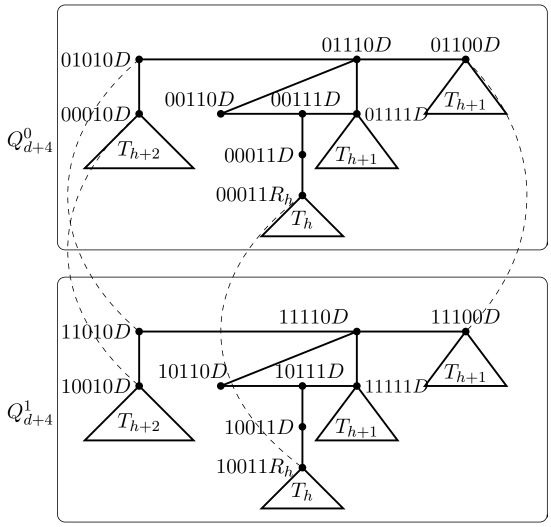

Figure 3(a). $i\tau_{h+2} \hookrightarrow Q^i_{d+4}, i \in \{0, 1\}$

Embedding:

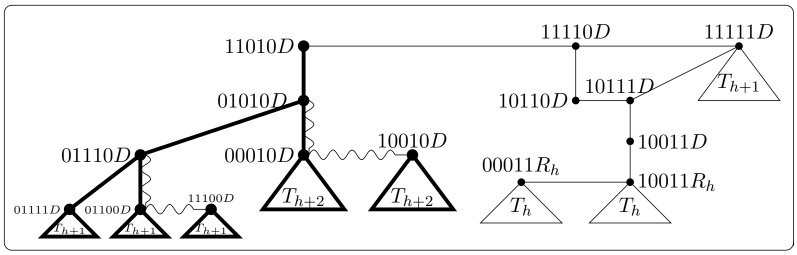

Figure 3(b). The edges $(01110D, 01100D)$ and $(01010D, 00010D)$ have edge congestion 2.

Output:

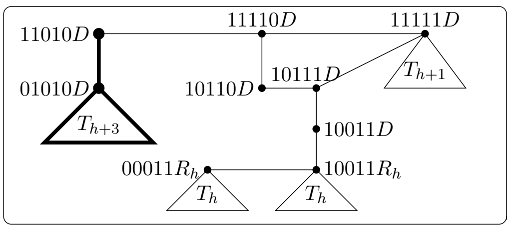

Figure 3(c). $\tau_{h+3} \hookrightarrow Q_{d+5}$. The subgraph shown on the right with light edges contains unutilized vertices.

Figure 3. The steps involved in an embedding of $\tau_{h+3} \hookrightarrow Q_{d+5}$, with the input $\tau_{h+2} \hookrightarrow Q_{d+4}$.

Here, we consider the decomposition of Q_{d+5} into two copies of Q_{d+4} and combine the embeddings such that we obtain an embedding of τ_{h+3} with $11010D$ as the deep root and $01010D$ as the root; see Figure 3(b). The three children c_l, c_m and c_r of the root R_{h+3} are mapped respectively, on to $01110D$, $00010D$ and $10010D$. The edge $R_{h+3}D \in \tau_{h+3}$ (mapped onto the edge $(11010D, 01010D)$) has dilation 1 and the edge $(11010D, 01010D)$ has congestion 1. Each of the parameters namely, dilation, edge congestion and node congestion of the embedding is bounded by 2.

Figure 3(c), also show embeddings of a T_{h+1} (with $11111D$ as the root) and two copies of T_h rooted at $00011R_h$ and $10011R_h$ whose vertices will be utilized in the subsequent steps. In all the embeddings, the edge joining the root and the deep root of the tree are mapped onto the adjacent nodes and they receive edge congestion 1. Every parameter dilation, node congestion and edge congestion of these embeddings is bounded by 2.

Here again, the output of this step, shown in Figure 3(c), is the input for the next step to embed τ_{h+4} in Q_{d+7}.

<u>**(iv)** $\boldsymbol{\tau_{h+4} \hookrightarrow Q_{d+7}}$</u> : The embedding (f, ρ) of τ_{h+4} in Q_{d+7} is described in Figure 4.

Similar to steps (i) and (ii), we obtain an embedding of τ_{h+4} with the deep root mapped on to $1011010D$ and the root of the tree T_{h+4} mapped on to $0011010D$; refer to Figure 4(b). The edges $R_{h+4}c_l$ and $R_{h+4}c_r$ have dilation 2. The edges $R_{h+4}c_m$ and $R_{h+4}D$ have dilation 1 and rest of the edges of T_{h+4} retain their dilation attained in the embedding τ_{h+3} in Q_{d+5}. Also, the edge $(1011010D, 0011010D)$ receives congestion 1. Each of the parameters namely, node congestion and edge congestion of the embedding is bounded by 2.

Figure 4(b) also shows two copies of T_{h+2} rooted at $1110111D$ and $0110111D$, and a T_{h+3} rooted at $1101010D$ embedded in Q_{d+7}. In all embeddings, the edge joining the root and the deep root of the tree are mapped onto the adjacent nodes and they receive edge congestion 1. The vertices of these trees are not yet utilized. These vertices will be utilized in the next step. The nodes $0010111D$ and $1010111D$ receive congestion 2; see Figure 2(b). Every parameter namely, dilation, node congestion and edge congestion of these embeddings is bounded by 2.

The output of this step, shown in Figure 4(c), is the input to embed τ_{h+5} in Q_{d+8} in the next step.

Input: The output of the embedding $\tau_{h+3} \hookrightarrow Q_{d+5}$; see Figure 3(c).

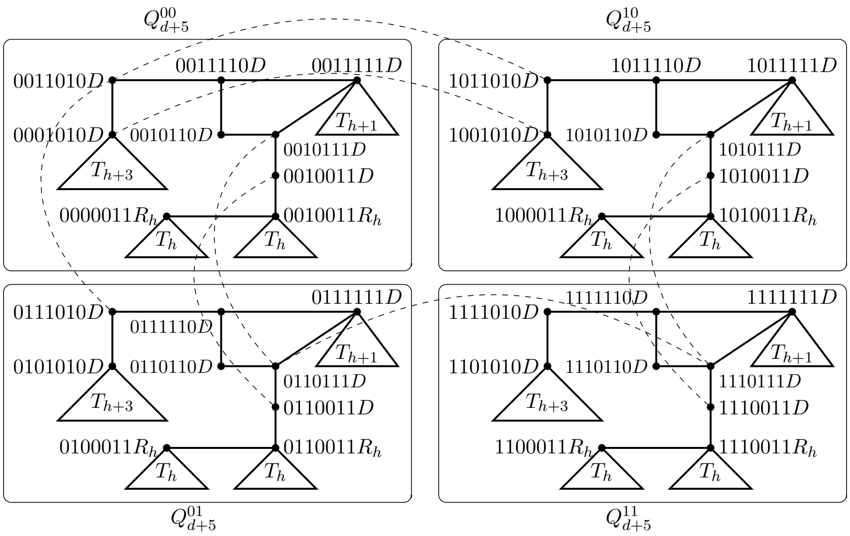

Figure 4(a). $ij\tau_{h+3} \hookrightarrow Q_{d+5}^{ij}, i, j \in \{0, 1\}$

Embedding:

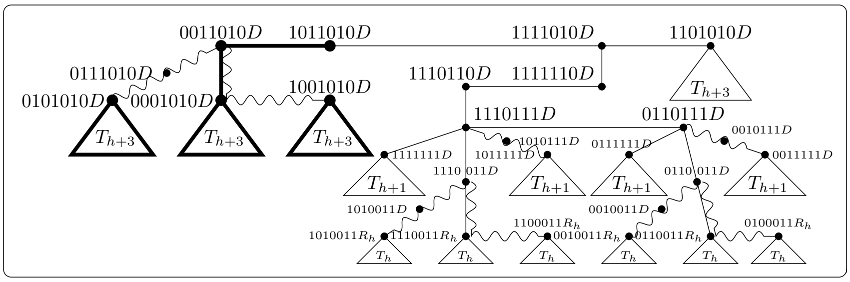

Figure 4(b). The edges $(0011010D, 0001010D)$,$(1110011D, 1110011R_h)$ and $(0110011D, 0110011R_h)$ have edge congestion 2.

Output:

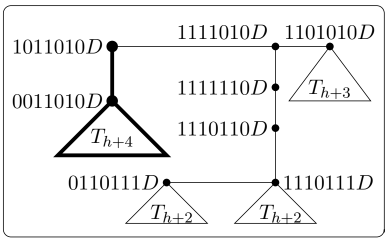

Figure 4(c). $\tau_{h+4} \hookrightarrow Q_{d+7}$. The subgraph shown on the right with light edges contains unutilized vertices.

Figure 4. The steps involved in an embedding of $\tau_{h+4} \hookrightarrow Q_{d+7}$, with the input $\tau_{h+3} \hookrightarrow Q_{d+5}$.

(v) $\tau_{\mathbf{h+5}} \hookrightarrow \mathbf{Q_{d+8}}$: The embedding (f, ρ) of τ_{h+5} in Q_{d+8} is described in Figure 5.

We decompose Q_{d+8} into two hypercubes Q^0_{d+7} and Q^1_{d+7}. And embed $0\tau_{h+4}$ in Q^0_{d+7} and $1\tau_{h+4}$ in Q^1_{d+7}; see Figure 5(a). We combine these two trees, as shown in Figures 5(a) and 5(b), to obtain the required embedding of τ_{h+5} into Q_{d+8} with $11011010D$ as the deep root and $01011010D$ as the root of the tree T_{h+5}. Note that, the edge $R_{h+5}D$ is mapped onto the edge $(01011010D, 11011010D)$. Therefore, $R_{h+5}D$ receives dilation 1 and moreover, $(01011010D, 11011010D)$ has edge congestion 1. The dilation of each of the edges of T_{h+5} is bounded by 2 and that the congestion of each node and edge of Q_{d+8} is also bounded by 2.

Hence, given $\tau_h \hookrightarrow Q_d$, we have obtained an embedding of τ_{h+5} into Q_{d+8} with load 1, dilation 2, node congestion 2 and edge congestion 2 in five steps (i) to (v). ■

Theorem 2.2. *Any complete ternary tree T_h is embeddable with load* 1, *dilation* 2, *node congestion* 2 *and edge congestion* 2 *into $Q_{d(h)}$, where*

$$d(h) = \begin{cases} \lceil (1.6)h \rceil, & \text{if } h \equiv 2 \pmod 5 \text{ or } h \equiv 4 \pmod 5, \\ \lceil (1.6)h \rceil + 1, & \text{if } h \equiv 0 \pmod 5 \text{ or } h \equiv 1 \pmod 5 \text{ or } h \equiv 3 \pmod 5. \end{cases}$$

Proof. We embed τ_h into $Q_{d(h)}$ by induction on $h \pmod 5$. The theorem follows, since T_h is a subtree of τ_h. For the base case, we have constructed embeddings of τ_0, τ_1, τ_2, τ_3 and τ_4 into Q_1, Q_3, Q_4, Q_6 and Q_7 respectively, with load 1, dilation 2, node congestion 2 and edge congestion 2. For the inductive step, we assume that $\tau_h \hookrightarrow Q_{d(h)}$ and show that $\tau_{h+5} \hookrightarrow Q_{d(h+5)}$, where $d(h+5) = \lceil (1.6)(h+5) \rceil$ or $\lceil (1.6)(h+5) \rceil + 1$. By Theorem 2.1, we have $\tau_{h+5} \hookrightarrow Q_{d(h)+8}$, where

$$\begin{aligned} d(h) + 8 &= (\lceil (1.6)h \rceil \text{ or } \lceil (1.6)h \rceil + 1) + 8 \\ &= (\lceil (1.6)h + 8 \rceil \text{ or } \lceil (1.6)h + 8 \rceil + 1) \\ &= \lceil (1.6)(h+5) \rceil \text{ or } \lceil (1.6)(h+5) \rceil + 1 \\ &= d(h+5). \end{aligned}$$

Since, in the basic step, we have $\tau_h \hookrightarrow Q_{\lceil (1.6)h \rceil}$, when $h = 2$ or 4, and we have $\tau_h \hookrightarrow Q_{\lceil (1.6)h \rceil + 1}$, when $h = 0, 1$ or 3, the theorem follows. ■

Input: The output of the embedding $\tau_{h+4} \hookrightarrow Q_{d+7}$; see Figure 4(c).

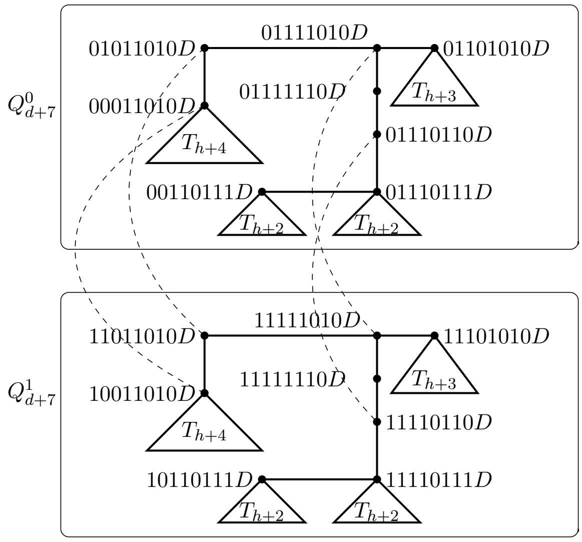

Figure 5(a). $i\tau_{h+4} \hookrightarrow Q^i_{d+7}, i \in \{0,1\}$

Embedding:

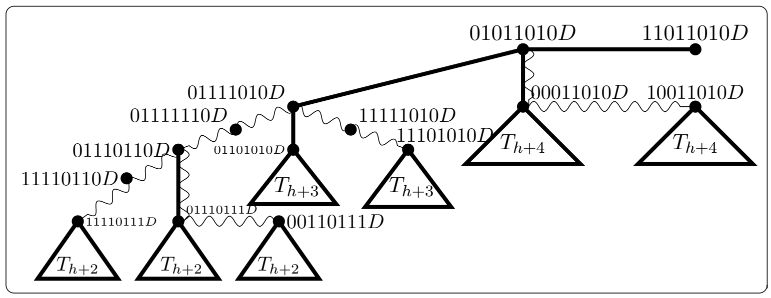

Figure 5(b). The edges $(01110110D, 01110111D)$ and $(01011010D, 00011010D)$ have edge congestion 2.

Output:

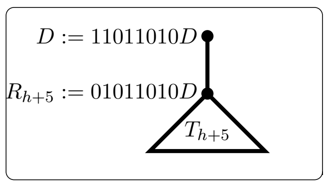

Figure 5(c). $\tau_{h+5} \hookrightarrow Q_{d+8}$

Figure 5. The steps involved in an embedding of $\tau_{h+5} \hookrightarrow Q_{d+8}$, with the input $\tau_{h+4} \hookrightarrow Q_{d+7}$.

Conclusions and Remarks

1. In this paper, we have obtained a load 1, dilation 2, node congestion 2 and edge congestion 2 embedding of the complete ternary tree of height h into a hypercube of dimension $d(h) = \lceil (1.6)h \rceil$ or $\lceil (1.6)h \rceil + 1$. More precisely,

$$d(h) = \begin{cases} \lceil (1.6)h \rceil, & \text{if } h \equiv 2 \,(\mathrm{mod}\ 5) \text{ or } h \equiv 4 \,(\mathrm{mod}\ 5), \\ \lceil (1.6)h \rceil + 1, & \text{if } h \equiv 0 \,(\mathrm{mod}\ 5) \text{ or } h \equiv 1 \,(\mathrm{mod}\ 5) \text{ or } h \equiv 3 \,(\mathrm{mod}\ 5). \end{cases}$$

Though the hypercube $Q_{d(h)}$ is not optimal, its dimension is very close to the dimension of the optimal hypercube which is $\lceil (\log_2 3)h \rceil (= \lceil (1.585)h \rceil)$ or $\lceil (\log_2 3)h \rceil + 1$.

2. Let $n(h)$ denote the dimension of the optimal hypercube of T_h. Let $d(h)$ be as defined in Theorem 1.3. We have computationally verified that $d(h) = n(h)$ for $2 \leq h \leq 15$, and that $n(h) \leq d(h) \leq n(h)+1$ for $16 \leq h \leq 80$. Therefore, by using our embeddings $\tau_0 \hookrightarrow Q_1$, $\tau_1 \hookrightarrow Q_3$, $\tau_2 \hookrightarrow Q_4$, $\tau_3 \hookrightarrow Q_6$ and $\tau_4 \hookrightarrow Q_7$ and the inductive description of the embedding given in the proof of Theorem 2.1, we conclude that

(i) for $2 \leq h \leq 15$, we have embedded T_h in its optimal hypercube, and

(ii) for $16 \leq h \leq 80$, we have embedded T_h either in its optimal hypercube or next-to-optimal hypercube.

Acknowledgement

The authors thank the referees for their valuable suggestions which improved our presentation.

REFERENCES

[1] S.L. Bezrukov, *Embedding complete trees into the hypercube*, Discrete Math. **110** (2001) 101–119.

[2] A.K. Gupta, D. Nelson and H. Wang, *Efficient embeddings of ternary trees into hypercubes*, Journal of Parallel and Distributed Computing **63** (2003) 619–629.

[3] I. Havel, *On certain trees in hypercube*, in: R. Bodendick and R. Henn, eds, Topics in Combinatorics and Graph Theory (Physica-Verlag, Heidelberg, 1990) 353–358.

[4] X.J. Shen, Q. Hu and W.F. Liang, *Embedding k-ary complete trees into hypercubes*, Journal of Parallel and Distributed Computing **24** (1995) 100–106.

[5] F.T. Leighton, Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes (Morgan Kauffmann, San Mateo, CA, 1992).

[6] B. Monien and H. Sudbourough, *Embedding one interconnection network into another*, Computing Supplementary **7** (1990) 257–282.

[7] J. Trdlička and P. Tvrdík, *Embedding of k-ary complete trees into hypercubes with uniform load*, Journal of Parallel and Distributed Computing **52** (1998) 120–131.

[8] A.Y. Wu, *Embedding of tree networks in hypercube*, Journal of Parallel and Distributed Computing **2** (1985) 238–249.

Received 24 October 2007
Revised 12 May 2008
Accepted 12 May 2008