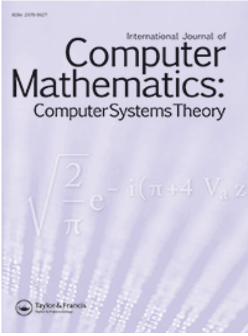


Taylor & Francis
Taylor & Francis Group

International Journal of Computer Mathematics: Computer Systems Theory

ISSN: 2379-9927 (Print) 2379-9935 (Online) Journal homepage: http://www.tandfonline.com/loi/tcom20

Certain height-balanced subtrees of hypercubes

Indhumathi Raman

INTERNATIONAL JOURNAL OF COMPUTER MATHEMATICS: COMPUTER SYSTEMS THEORY, 2016
http://dx.doi.org/10.1080/23799927.2016.1181671

Taylor & Francis
Taylor & Francis Group

Certain height-balanced subtrees of hypercubes

Indhumathi Raman

School of Information Technology and Engineering, VIT University, Vellore, India

ABSTRACT

A height-balanced tree is a desired data structure for performing operations such as search, insert and delete, on high-dimensional external data storage. Its preference is due to the fact that it always maintains logarithmic height even in worst cases. It is a rooted binary tree in which for every vertex the difference (denoted as balance factor) in the heights of the subtrees, rooted at the left and the right child of the vertex, is at most one. In this paper, we consider two subclasses of height-balanced trees $\mathcal{X}$ and $\mathcal{Y}$. A tree in $\mathcal{X}$ is such that all the vertices up to (a predetermined) level t has balance factor one and the remaining vertices have balance factor zero. A tree in $\mathcal{Y}$ is such that all the vertices at alternate levels up to t has balance factor one and the remaining vertices have balance factor zero. We prove that every tree in the classes $\mathcal{X}$ and $\mathcal{Y}$ is a subtree of the hypercube.

ARTICLE HISTORY
Received 2 November 2015
Revised 18 March 2016
Accepted 24 March 2016

KEYWORDS
Height-balanced tree; embedding; subgraph; hypercube

AMS SUBJECT CLASSIFICATION
05C78; 05C85; 68R10

1. Introduction

1.1. Embedding

The problem of efficiently implementing parallel algorithms on parallel computers has been studied as a graph embedding problem. Let the computational structure of a parallel algorithm A be represented by a graph G_A and the interconnection network of a parallel computer N be represented by a graph H_N. An embedding of G_A into H_N describes the working of the parallel algorithm A when implemented on N. A formal definition of embedding is as follows:

Definition 1.1: *An embedding ϕ of a graph $G = (V_G, E_G)$ into a graph $H = (V_H, E_H)$ is an injection $\phi : V_G \rightarrow V_H$ (not necessarily onto) such that every edge $(u, v) \in E_G$ is mapped to a shortest path connecting $\phi(u)$ and $\phi(v)$ in* H.

An extensive survey of embedding various graphs into interconnection networks such as hypercubes, meshes and tori is given in [2–4, 14, 16]. Two of the parameters that measure the quality of an embedding ϕ are dilation and expansion.

- The dilation $\mathrm{dil}(\phi)$ of an embedding ϕ is defined as, $\max\{\mathrm{dil}(u, v) : (u, v) \in E_G\}$ where $\mathrm{dil}(u, v)$ is defined to be the length of a shortest path connecting $\phi(u)$ and $\phi(v)$ in H for an edge $(u, v) \in E_G$.
- The expansion(ϕ) is defined to be the ratio $|V_H|/|V_G|$.

The $\mathrm{dil}(\phi)$ is used to estimate the computational running time and the expansion(ϕ) is used to estimate the number of unutilized processors in N. If $\mathrm{dil}(\phi) = 1$, then G is isomorphic to (denoted by $\simeq$) a subgraph of H; we write $G \subseteq H$. Additionally if expansion$(\phi) = 1$, then G is a spanning

CONTACT Indhumathi Raman indhumathi.r@vit.ac.in

subgraph of H. An embedding with dilation 1 and expansion 1 is the one with least communication delay and the most cost effective and hence is desirable. However for most embedding problems, it is very difficult to obtain an embedding that minimizes these two parameters simultaneously. Therefore, some trade-offs among these parameters must be made.

1.2. Hypercubes

Among the interconnection networks of parallel computers, the binary *hypercube* has received much attention. An important property of the hypercube, which makes it popular, is its ability to efficiently simulate the message routings of other interconnection networks. A formal definition of the hypercube is stated here.

Definition 1.2: *An n-dimension hypercube, Q_n, has 2^n vertices each labelled with a binary string of length n. Two vertices are adjacent if and only if their labels differ in exactly one position.*

A graph G is said to be t-arc transitive if for any two paths on $t+1$ vertices $P = (u_0, u_1, \ldots, u_t)$ and $Q = (v_0, v_1, \ldots, v_t)$, there exists an automorphism α of G such that $\alpha(u_i) = v_i$, for every $i, 0 \le i \le t$. The 0-arc transitive and 1-arc transitive are referred to as vertex-transitive and edge-transitive in this paper. It is well known that Q_n is a t-arc transitive graph for $t = 0, 1, 2$ (refer [13]).

If G is a bipartite graph with $V(G) = [X, Y]$ such that (i) $2^{n-1} < |V(G)| \le 2^n$ and (ii) $|X|, |Y| \le 2^{n-1}$, then G is said to be balanced. Also, Q_n is called the *optimal hypercube* of G and n is called the *optimal dimension* of G. It is desirable to embed G into a hypercube of dimension n or at least close to n so as to minimize the expansion of the embedding. It is obvious that the balancedness of G is a necessary condition for G to be a subgraph of Q_n. However, it is not sufficient. For example, the star graph $K_{1,4}$ is a balanced bipartite graph with $|V(G)| = 5 < 2^3$ and $|X|, |Y| \le 4$, but $K_{1,4}$ is not a subtree of Q_3 since its maximum vertex degree is 4 but Q_3 is 3-regular.

1.3. Height-balanced trees

Trees generally form the underlying data structure for several parallel algorithms that employ divide-and-conquer rule, branch-and-bound technique and so on. A binary search tree, or BST, is a binary tree whose vertices are arranged such that for every vertex v, all the vertices in its left subtree have a value less than v, and all the vertices in its right subtree have a value less than v. In the average case, BSTs offer logarithmic time for inserting/deleting/searching a value. The disadvantage of BSTs is that, in the worst case, their running time is linear. This happens if the items are inserted into the BST are in order. In such a case, a BST performs no better than an array. However, there exist self-balancing BSTs which ensures that, regardless of the order of the data inserted, the tree always maintains a logarithmic running time. Certain examples of self-balancing BSTs are red-black trees, 2–3–4 trees, Splay trees, B-trees, Adelson, Velskii and Landis (AVL) trees, etc. The computational efficiency of AVL trees has motivated a vast study in information storage and retrieval and to implement parallel dictionary routines, see [9, 15]. The fact that the worst height of a height-balanced tree is logarithmic (see [1, 12]) has motivated the author to study the embedding of height-balanced trees into hypercubes.

An AVL tree is also called a height-balanced tree and is formally defined in [1] as below. Figure 1(a) shows a few height-balanced trees of heights 1,2,3 (the height of a tree is taken as the length of a longest path from the root), respectively.

Definition 1.3: *A rooted binary tree is said to be height-balanced if for every vertex v, the heights of the subtrees rooted at the left and right child of v, differ by at most one.*

For a vertex v in a height-balanced tree T, the balance factor $\mathbf{b}_T(v)$ of v is defined as the difference between the heights of left and right subtrees of v. It is usually assumed that the height of an empty

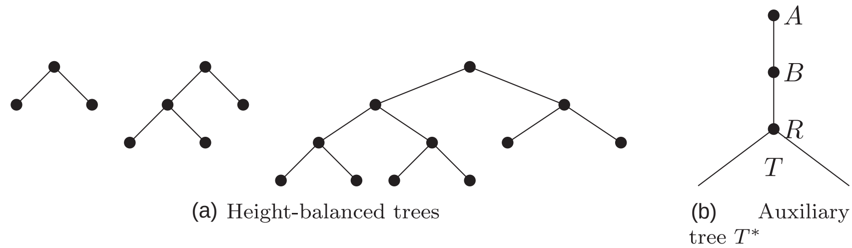


Figure 1. Examples of height-balanced trees and auxiliary tree: (a) height-balanced trees and (b) auxiliary tree T^*.

tree is -1 and consequently, (i) for a leaf $l \in V(T)$, $\mathbf{b}_T(l) = 0$ and (ii) the tree consisting of the root alone has height 0. Clearly, $\mathbf{b}_T(v) = 0$ for every $v \in V(T)$ if and only if T is a complete binary tree.

A tree T is a subtree of a hypercube Q_n if and only if T admits a labelling of its vertices using n-bit binary strings such that the labels of the adjacent vertices differ in exactly one position. Hence an embedding $L : T \to Q_n$ amounts to a labelling of the vertices of T with binary strings of length n. To achieve our embeddings, we make use of a small extension in the structure of a tree T with root R: we add two new vertices A, B and two new edges (A, B) and (B, R). We call this supertree of T as the *auxiliary tree* of T (denoted as T^*) and call the path (A, B, R) as the *auxiliary path* of T^*. Figure 1(b) shows the auxiliary tree of a binary tree T.

1.4. Outline of the paper

In 1984, Havel [10] conjectured that *any binary tree can be embedded into its optimal hypercube with dilation 2.* There are several papers which show that the conjecture is true for special classes of binary trees (see [5, 8, 17]). The following are some of the well-known results on embedding height-balanced trees into hypercube.

Proposition 1.4 ([3]): *For all $h \geq 3$, the complete binary tree C_h of height* h *is embeddable into its optimal hypercube Q_{h+1} with dilation 2. However, C_h is a subtree of its next-to-optimal hypercube Q_{h+2}.*

Proposition 1.5 ([6]): *For all $h \geq 1$, the Fibonacci tree $\mathbb{F}_h$ of height* h *(a height-balanced tree in which the balance factor of all the non-leaf vertices is one) is a subtree of $Q_{\lceil 0.75h \rceil + 1}$.*

Proposition 1.6 ([7]): *Let HBT_h be a class of height-balanced trees in which the balance factor is arbitrary in the first three levels and is zero thereafter and let HBT_h^* be the class of the corresponding auxiliary trees. For all $h \geq 1$, every balanced height-balanced tree of height* h *in HBT_h^* is a subtree of its optimal hypercube.*

In [11], the authors have considered two classes of height-balanced trees that are recursive in structure and have proved that every tree in these two classes is a subtree of the hypercube. In this paper, we continue to embed other height-balanced trees and in our embeddings, we have reduced the dilation from 2 (as stated in the conjecture) to 1 but at the cost of increasing the expansion. The gap between the obtained expansion and the minimum expansion (as demanded in the conjecture) is $O(h)$. In Section 2, we define the two subclasses of height-balanced trees and also find the optimal dimension of the hypercube into which the trees of these subclasses can be embedded. In Section 3, we embed every tree in the two subclasses into the hypercube.

2. Preliminaries

In this section, we define two subclasses of height-balanced trees $\mathcal{X}$ and $\mathcal{Y}$ and determine the dimension of the optimal hypercube into which every tree of these subclasses can be embedded. If T_h

is a height-balanced tree of height h, we denote its levels by $L_0, L_1, \dots L_h$, where for $0 \le i \le h$, $L_i = \{v \in V(T_h) : \mathrm{dist}_{T_h}(v, \mathrm{root}) = i\}$.

Definition 2.1: *Let $X_{h,t}$ be a height-balanced tree of height h such that*

$$\mathbf{b}_{X_{h,t}}(v) = \begin{cases} 1 & \text{if } v \in \bigcup_{i=0}^{t-1} L_i, \\ 0 & \text{if } v \in \bigcup_{i=t}^{h} L_i. \end{cases}$$

We denote by $\mathcal{X}$ the class $\{X_{h,t} : h \ge 0, 0 \le t \le h/2\}$ of trees.

Definition 2.2: *Let $Y_{h,t}$ be a height-balanced tree of height h such that*

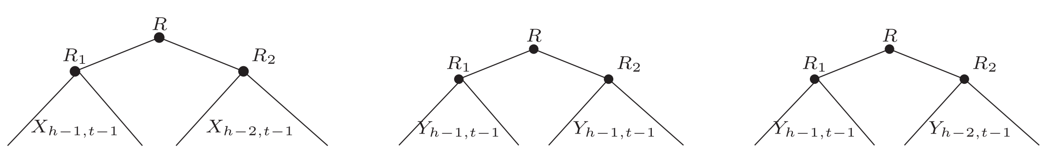
$$\mathbf{b}_{Y_{h,t}}(v) = \begin{cases} 0 & \text{if } v \in L_0 \cup L_2 \cup \cdots \cup L_{t-2}, & t \text{ is even}, \\ 1 & \text{if } v \in L_1 \cup L_3 \cup \cdots \cup L_{t-1}, & t \text{ is even}, \\ 1 & \text{if } v \in L_0 \cup L_2 \cup \cdots \cup L_{t-1}, & t \text{ is odd}, \\ 0 & \text{if } v \in L_1 \cup L_3 \cup \cdots \cup L_{t-2}, & t \text{ is odd}, \\ 0 & \text{if } v \in \bigcup_{i=t}^{h} L_i. \end{cases}$$

We denote by $\mathcal{Y}$ the class $\{Y_{h,t} : h \ge 0,\ 0 \le t \le h/2\}$ of trees.

The balance condition imposed on the vertices of the trees $X_{h,t}$ and $Y_{h,t}$ may seem hard. The motivation for defining the class $\mathcal{X}$ of trees is as follows: the complete binary tree (where the balance factor of all vertices is zero) and the Fibonacci tree (where the balance factor of all non-leaf vertices is one) are embeddable into the hypercube (see Propositions 1.4 and 1.5). An obvious question that arises is: *Is a midway tree (half the vertices with balance factor one and other half the vertices until level* t *with balance factor zero) embeddable into hypercube?* Such a midway tree is $X_{h,t}$. By further making the balance factor of every vertex in the first t alternate levels of $X_{h,t}$ to zero, we get $Y_{h,t}$. The trees in the classes $\mathcal{X}$ and $\mathcal{Y}$ can be defined recursively also.

Definition 2.3: *$X_{h,0} := C_h$, and for $1 \le t \le \lfloor h/2 \rfloor$, $X_{h,t}$ is formed by taking one copy of $X_{h-1,t-1}$ with root R_1, one copy of $X_{h-2,t-1}$ with root R_2, a new vertex R and adding the edges (R, R_1) and (R, R_2); refer Figure 2(a).*

Definition 2.4: *$Y_{h,0} := C_h$, for $1 \le t \le \lfloor h/2 \rfloor$, if t is even, $Y_{h,t}$ is formed by taking two copies of $Y_{h-1,t-1}$ with roots R_1 and R_2, a new vertex R and adding the edges (R, R_1) and (R, R_2) (refer*



(a) Recursive definition of $X_{h,t}$ for $t \ge 1$ (b) Recursive definition of $Y_{h,t}$ when $t \ge 1$ is even (c) Recursive definition of $Y_{h,t}$ when $t \ge 1$ is odd

Figure 2. Structure of $X_{h,t}$ and $Y_{h,t}$ for $1 \le t \le \lfloor h/2 \rfloor$: (a) $X_{h,t}$ for $t \ge 1$, (b) $Y_{h,t}$ when $t \ge 1$ is even and (c) $Y_{h,t}$ when $t \ge 1$ is odd.

Figure 2(b)) and if t is odd, $Y_{h,t}$ is formed by taking one copy of $Y_{h-1,t-1}$ with root R_1, one copy of $Y_{h-2,t-1}$ with root R_2, a new vertex R *and adding the edges* (R, R_1) *and* (R, R_2) *(refer Figure 2(c)).*

From Definitions 2.1 through 2.4, the following proposition is obvious.

Proposition 2.5: *For any two integers $t_1 \geq t_2 > 0$, the following hold:*

- $X_{h,t_1} \subseteq X_{h,t_2}$
- $Y_{h,2t_1} \subseteq Y_{h,2t_2}$
- $Y_{h,2t_1-1} \subseteq Y_{h,2t_2-1}$

One may see that $\mathbf{b}_{X_{h,t}}(v) = 0$ and $\mathbf{b}_{Y_{h,t}}(v) = 0$ for all vertices $v \in \bigcup_{i=t}^{h} L_i$ of $X_{h,t}$ and $Y_{h,t}$. This implies that the subtree of $X_{h,t}(Y_{h,t})$ rooted at a vertex in L_t is a complete binary tree. A precise value for the number of vertices of the trees is given in the next proposition. This helps us to determine the optimal dimensions of $X_{h,t}$ and $Y_{h,t}$.

Proposition 2.6: *For every* t, $0 \leq t \leq \lfloor h/2 \rfloor$,

- $|V(X_{h,t})| = (2^{h-2t+1})(3^t) - 1.$
- $|V(Y_{h,t})| = \begin{cases} -1 + (2^{h-t+1})(3^{t/2}) & \text{if } t \text{ is even,} \\ -1 + (2^{h-t})(3^{(t+1)/2}) & \text{if } t \text{ is odd.} \end{cases}$

Proof: We prove by induction on t. For $t=0$, we have $X_{h,0} \simeq C_h$ and hence $x_{h,t} = |V(C_h)| = 2^{h+1} - 1$. Let $x_{h,t} = |V(X_{h,t})|$ and by induction hypothesis, $x_{h-1,t-1} = (2^{h-2t+2})(3^{t-1}) - 1$ and $x_{h-2,t-1} = (2^{h-2t+1})(3^{t-1}) - 1$. From Definition 2.3, we have $x_{h,t} = x_{h-1,t-1} + x_{h-2,t-1} + 1 = (2^{h-2t+2})(3^{t-1}) + (2^{h-2t+1})(3^{t-1}) - 1 = (2^{h-2t+1})(3^t) - 1$. The number of vertices of $Y_{h,t}$ can be proved on similar lines. ■

Proposition 2.7: *The optimal dimensions of $X_{h,t}$ and $Y_{h,t}$ are $h - \lfloor 0.42t \rfloor + 1$ and $h - \lfloor 0.21t \rfloor + 1$, respectively.*

Proof: The optimal dimension of $X_{h,t}$ is $\lceil \log_2(|V(X_{h,t})|) \rceil$ which equals $\lceil h - 2t + 1 + 1.58t \rceil$ (we use Proposition 2.6 and $3 \sim 2^{1.58}$) $= h - \lfloor 0.42t \rfloor + 1$. Similarly, the optimal dimension of $Y_{h,t}$ is $\lceil \log_2(|V(Y_{h,t})|) \rceil$ which equals $\lceil h - t + 1.58((t+1)/2) \rceil = \lceil h - t + 0.79t + 0.79 \rceil = h - \lfloor 0.21t \rfloor + 1$. ■

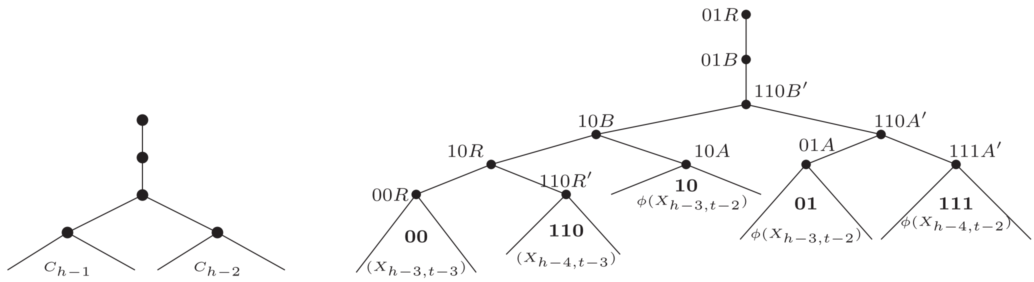
3. Embedding $X_{h,t}$ and $Y_{h,t}$ into hypercube

In this section, we embed every tree in the classes $\mathcal{X}$ and $\mathcal{Y}$ into hypercube. Specifically, we prove that

- Every $X_{h,t} \in \mathcal{X}$ is embeddable into $Q_{m(h,t)}$ with dilation 1 where $m(h,t) = h - \lfloor t/3 \rfloor + 1$.
- Every $Y_{h,t} \in \mathcal{Y}$ is embeddable into $Q_{d(h,t)}$ with dilation 1 where $d(h,t) = h - \lfloor t/6 \rfloor + 2$.

We note that the gap between the dimension of the hypercube (into which the trees are embedded) and the optimal dimension of the trees (given in Proposition 2.7) is $O(t) = O(h)$ since $t \leq h/2$. The problem of embedding the trees $X_{h,t}$ and $Y_{h,t}$ into their optimal hypercube is left open.

Theorem 3.1: *Let $X_{h,t}$ be the height-balanced tree as defined in Definition 2.1. For every $h \geq 1$ and every t, $1 \leq t \leq \lfloor h/2 \rfloor$, $X^*_{h,t} \subseteq Q_{m(h,t)}$ where $m(h,t) = h - \lfloor t/3 \rfloor + 1$.*



(a) The tree $B^*(h) \subseteq Q_{h+1}$

(b) Labelling of $X^*_{h,t}$: Here $R = 0^{r-2}$, $B = 0^{r-3}1$, $A = 0^{r-4}11$, $R' = 0^{r-3}$, $B' = 0^{r-4}1$, $A' = 0^{r-5}11$

Figure 3. Embedding of $X^*_{h,t}$ into Q_r: (a) the tree $B^*(h) \subseteq Q_{h+1}$ and (b) labelling of $X^*_{h,t}$: here $R = 0^{r-2}$, $B = 0^{r-3}1$, $A = 0^{r-4}11$, $R' = 0^{r-3}$, $B' = 0^{r-4}1$, $A' = 0^{r-5}11$.

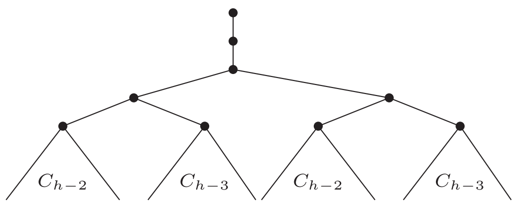
Proof: The idea of our proof is to label the vertices of $X^*_{h,t}$ using $m(h,t)$-bit binary strings where $m(h,t) = h - \lfloor t/3 \rfloor + 1$, by induction on t. For the base case, we have $X^*_{h,2} \subseteq X^*_{h,1} \simeq B^*(h) \subseteq Q_{h+1}$; the first containment follows from Proposition 2.5, the tree $B^*(h)$ is shown in Figure 3(a) and the last containment follows from the fact that $B^*(h) \subseteq HBT^*_h$ and from Proposition 1.6. The tree $X^*_{h,3} \subseteq HBT^*_h \subseteq Q_h$ (the first containment follows from the definitions of $X^*_{h,3}$ and HBT^*_h and the second containment follows from Proposition 1.6). Hence the theorem is true for initial values of $t = 1,2,3$. Further for $t \geq 4$, without loss of generality, we prove the theorem such that the auxiliary path (A, B, R) of $X^*_{h,t}$ is mapped to $(0^{m(h,t)-2}11, 0^{m(h,t)-1}1, 0^{m(h,t)})$. We call this condition on the auxiliary path as *auxiliary condition*.

Since $X^*_{h,3k+2} \subseteq X^*_{h,3k+1} \subseteq X^*_{h,3k}$ (by Proposition 2.5) and $m(h, 3k+2) = m(h, 3k+1) = m(h, 3k)$, it is sufficient to prove that for $t = 3k$, $X^*_{h,t} \subseteq Q_r$ where $r = m(h, 3k)$ and this is shown below. We may verify that, $r = m(h,t) = m(h-3, t-3) + 2 = m(h-4, t-3) + 3 = m(h-3, t-2) + 2 = m(h-4, t-2) + 3$. The tree $X^*_{h,t}$ contains one copy of $X_{h-3,t-3}$, one copy of $X_{h-4,t-3}$, two copies of $X_{h-3,t-2}$ and one copy of $X_{h-4,t-2}$. By induction hypothesis, we are given embeddings $X^*_{h-3,t-3} \subseteq Q_{m(h-3,t-3)} \simeq Q_{r-2}$, $X^*_{h-4,t-3} \subseteq Q_{m(h-4,t-3)} \simeq Q_{r-3}$, $X^*_{h-3,t-2} \subseteq Q_{m(h-3,t-2)} \simeq Q_{r-2}$ and $X^*_{h-4,t-2} \subseteq Q_{m(h-4,t-2)} \simeq Q_{r-3}$ satisfying the auxiliary condition. Let (A, B, R) and (A', B', R') denote the auxiliary paths of $X^*_{h-3,\alpha}$ and $X^*_{h-4,\alpha}$, respectively, where $\alpha \in \{t-2, t-3\}$. By auxiliary condition, $R = 0^{r-2}$, $B = 0^{r-3}1$, $A = 0^{r-4}11$, $R' = 0^{r-3}$, $B' = 0^{r-4}1$, $A' = 0^{r-5}11$. Since Q_n $(n \in \{r-2, r-3\})$ is 2-arc transitive, there exists an automorphism ϕ of Q_n such that the path (A, B, R) is mapped onto the path (R, B, A) and the path (A', B', R') is mapped onto the path (R', B', A'). We label the vertices of $X^*_{h,t}$ as follows: (i) apply the automorphism ϕ on the vertices of $X^*_{h-3,t-2}$ and $X^*_{h-4,t-2}$ then prefix the labels of all the subtrees by a 2 or 3 bit label as shown in Figure 3(b). Label the vertices on the auxiliary path of $X^*_{h,t}$ with $110B'$, $01B$ and $01R$. Since Q_r is 2-arc transitive, there exists an automorphism of Q_r which maps the auxiliary path of $X^*_{h,t}$ onto $(0^{r-2}11, 0^{r-1}1, 0^r)$ thus satisfying the auxiliary condition. Since the labels of the vertices of $X^*_{h,t}$ are distinct, the provided labelling is a required embedding. ■

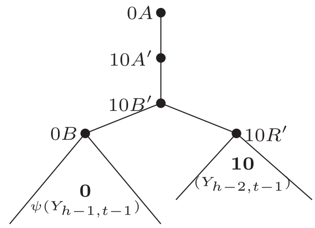
Before we proceed to embed $Y^*_{h,t}$ into $Q_{h-\lfloor t/6 \rfloor+2}$ with unit dilation, we prove the following lemma.

Lemma 3.2: *If $Y_{h,t}$ is a height-balanced tree as defined in Definition 2.2, then for $1 \leq t \leq 6$, $Y^*_{h,t} \subseteq Q_{h+1}$.*

Proof: The tree $Y^*_{h,1} \simeq B^*(h) \subseteq Q_{h+1}$ (the containment follows from Proposition 1.6 since $B^*(h) \subseteq HBT^*_h$). The tree $Y^*_{h,2} \simeq D^*(h) \subseteq Q_{h+1}$ (the tree $D^*(h)$ is shown in Figure 4(a) and the containment



(a) The tree $D^*(h) \subseteq Q_{h+1}$



(b) Labelling of $Y^*_{h,t}$ for odd t

Figure 4. Embedding of $Y^*_{h,t}$ into $Q_{d(h,t)}$ when t is odd: (a) the tree $D^*(h) \subseteq Q_{h+1}$ and (b) labelling of $Y^*_{h,t}$ for odd t.

follows from Proposition 1.6 since $D^*(h) \subseteq HBT^*_h$). The trees $Y^*_{h,5} \subseteq Y^*_{h,3} \subseteq Y^*_{h,1} \subseteq Q_{h+1}$ and the trees $Y^*_{h,6} \subseteq Y^*_{h,4} \subseteq Y^*_{h,2} \subseteq Q_{h+1}$. ■

Theorem 3.3: *Let $Y_{h,t}$ be a height-balanced tree as defined in Definition 2.2. For every $h \geq 1$ and every $t, 0 \leq t \leq \lfloor h/2 \rfloor$, $Y^*_{h,t} \subseteq Q_{d(h,t)}$ where $d(h,t) = h - \lfloor t/6 \rfloor + 2$.*

Proof: We prove the theorem by induction on t. For $t = 0$, the tree $Y^*_{h,0} \simeq C^*_h \subseteq Q_{h+2}$ (the containment follows from Proposition 1.4). For $1 \leq t \leq 5$, by Lemma 3.2, we have $Y^*_{h,t} \subseteq Q_{h+1} \subseteq Q_{h+2}$. For $t = 6$, the theorem follows from Lemma 3.2. Hence for the base case, the theorem is true for $0 \leq t \leq 6$. For $t \geq 7$, we prove that $Y^*_{h,t} \subseteq Q_{d(h,t)}$ such that the auxiliary path of $Y^*_{h,t}$ is mapped onto the path $(0^{d(h,t)-2}11, 0^{d(h,t)-1}1, 0^{d(h,t)})$ (recall that this is called auxiliary condition).

Case 1: t is odd

In this case, the balance factor of the root is 1 and $t \equiv 1, 3, 5 \pmod 6$. It can be easily verified that $q = d(h,t) = d(h-1, t-1) + 1 = d(h-2, t-1) + 2$. By induction hypothesis, there exist embeddings $Y^*_{h-1,t-1} \subseteq Q_{d(h-1,t-1)} \simeq Q_{q-1}$ and $Y^*_{h-2,t-1} \subseteq Q_{d(h-2,t-1)} \simeq Q_{q-2}$ satisfying the auxiliary condition. Let (A, B, R) and (A', B', R') denote the auxiliary paths of $Y^*_{h-1,t-1}$ and $Y^*_{h-2,t-1}$, respectively. By the auxiliary condition, $R = 0^{q-1}$, $B = 0^{q-2}1$, $A = 0^{q-3}11$, $R' = 0^{q-2}$, $B' = 0^{q-3}1$, $A' = 0^{q-4}11$. Since Q_{q-1} is edge-transitive, there exists an automorphism ψ of Q_{q-1} such that the edge (B, R) is mapped onto the edge (A, B). We label the vertices of $Y^*_{h,t}$ as follows: (i) apply the automorphism ψ on the labels of the vertices of $Y^*_{h-1,t-1}$ and then prefix the labels by 0, (ii) consider the labelling of $Y^*_{h-2,t-1}$ (by induction hypothesis) and prefix the labels by 10 and (iii) label the vertices on the auxiliary path of $Y^*_{h,t}$ with $10B'$, $10A'$ and $0A$. The above labelling technique is shown in Figure 4(b). Since Q_q is 2-arc transitive, there exists an automorphism of Q_q which maps the auxiliary path of $Y^*_{h,t}$ onto $(0^{q-2}11,\ 0^{q-1}1,\ 0^q)$, thus satisfying the auxiliary condition.

Case 2: t is even

In this case, the balance factor of the root is 0 and $t \equiv 0, 2, 4 \pmod 6$. Since $Y^*_{h,6k+4} \subseteq Y^*_{h,6k+2} \subseteq Y^*_{h,6k}$ (by Proposition 2.5) and $d(h, 6k+4) = d(h, 6k+2) = d(h, 6k)$, it is sufficient to prove that for $t = 6k$, $Y^*_{h,t} \subseteq Q_p$ where $p = d(h, 6k)$ and this is shown below.

It can be easily verified that $p = d(h,t) = d(h-1, t-1) = d(h-6, t-4) + 5 = d(h-6, t-5) + 5 = d(h-6, t-6) + 5 = d(h-7, t-6) + 6$. By induction hypothesis, there exist embeddings $Y^*_{h-6,t-6} \subseteq Q_{d(h-6,t-6)} \simeq Q_{p-5}$, $Y^*_{h-6,t-5} \subseteq Q_{d(h-6,t-5)} \simeq Q_{p-5}$, $Y^*_{h-6,t-4} \subseteq Q_{d(h-6,t-4)} \simeq Q_{p-5}$, $Y^*_{h-7,t-6} \subseteq Q_{d(h-7,t-6)} \simeq Q_{p-6}$. Let (A, B, R) be the auxiliary path of $Y^*_{h-6,\beta}$ where $\beta \in \{t-4, t-5, t-6\}$ and let (A', B', R') be the auxiliary path of $Y^*_{h-7,t-6}$, both satisfying the auxiliary condition. Hence $A = 0^{p-7}11$, $B = 0^{p-6}1$, $R = 0^{p-5}$, $A' = 0^{p-8}11$, $B' = 0^{p-7}1$, $R' = 0^{p-6}$. Let ϕ, ψ and θ be automorphisms of a hypercube Q_n where $n \in \{p-5, p-6\}$ which maps the path $(0^{n-2}11, 0^{n-1}1, 0^n)$ onto $(0^n, 0^{n-1}1, 0^{n-2}11)$, the edge $(0^{n-1}1, 0^n)$ onto the edge $(0^{n-2}11, 0^{n-1}1)$ and the edge $(0^{n-1}1, 0^n)$ onto the edge $(0^n, 0^{n-1}1)$ respectively. Such automorphisms exist since hypercube is 2-arc transitive and edge-transitive. Let π be the identity automorphism. We label the vertices of $Y^*_{h,t}$ with p-bit string labels as follows: consider a drawing of $Y^*_{h,t}$ drawn three levels down and this drawing contains four copies of $Y_{h-3,t-3}$ and two copies of $Y_{h-3,t-2}$ (refer to Figure 5). If we further redraw each copy of the trees $Y_{h-3,t-3}$ and $Y_{h-3,t-2}$ three levels down, the structure of $Y_{h-3,t-3}$ has two copies of $Y_{h-6,t-6}$, two copies of $Y_{h-7,t-6}$ and two copies of $Y_{h-6,t-5}$ (refer to Figure 6) and the structure of $Y_{h-3,t-2}$ has four copies of $Y_{h-6,t-5}$ and two copies of $Y_{h-6,t-4}$ (refer to Figure 10). Hence $Y^*_{h,t}$ has in total of $4(2+2+2)+2(4+2)=36$ subtrees when drawn $3+3=6$ levels down. The vertices of $Y^*_{h,t}$ at levels 0– 3 are labelled using A,B,R each prefixed by a 5-bit label. Four labellings of $Y_{h-3,t-3}$ are shown in Figures 6–9. In these figures, the labels of $Y_{h-6,t-6}$, $Y_{h-6,t-5}$ (and $Y_{h-7,t-6}$) are prefixed by a 5-bit (and 6-bit) label after being subject to one of the automorphism in $\{\pi, \phi, \psi, \theta\}$. Two labellings of $Y_{h-3,t-2}$ are shown in Figures 10 and 11. In these figures, the labels of $Y_{h-6,t-5}$ and $Y_{h-6,t-4}$ are prefixed by a 5-bit label after being subject to one of the automorphism in $\{\pi, \phi, \psi, \theta\}$. The above labelling technique of $Y^*_{h,t}$ is depicted in Figures 5 – 11 and we note that

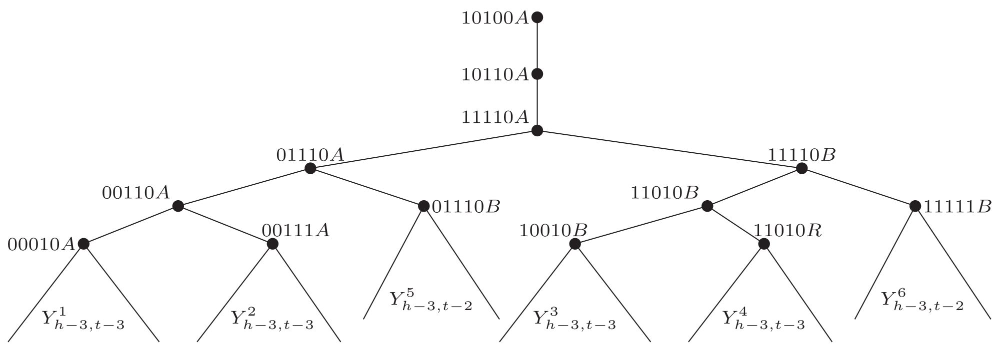


Figure 5. Embedding of $Y^*_{h,6k}$ into Q_p; the labelling of $Y^i_{h-3,t-3}$, for $1 \le i \le 4$, is given in Figures 6–9, respectively, and the labelling of $Y^i_{h-3,t-2}$, for $i=5,6$, is given in Figures 10 and 11, respectively. In Figures 5– 11, $A = 0^{p-7}11$, $B = 0^{p-6}1$, $R = 0^{p-5}$ and $A' = 0^{p-8}11$, $B' = 0^{p-7}1$, $R' = 0^{p-6}$.

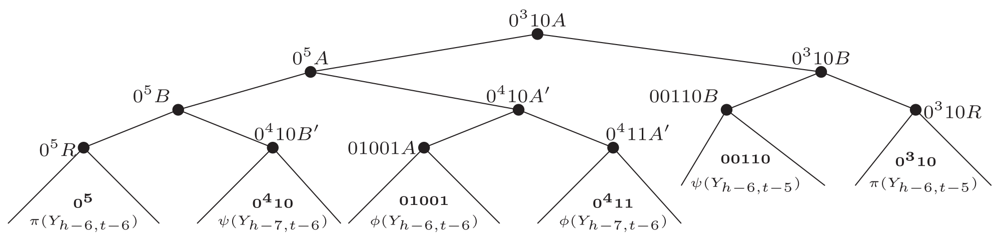


Figure 6. Labelling of $Y^1_{h-3,t-3}$: the embedding of $Y_{h-6,t-6}$ into Q_{p-5}, $Y_{h-7,t-6}$ into Q_{p-6} and $Y_{h-6,t-5}$ into Q_{p-5} follows by induction hypothesis.

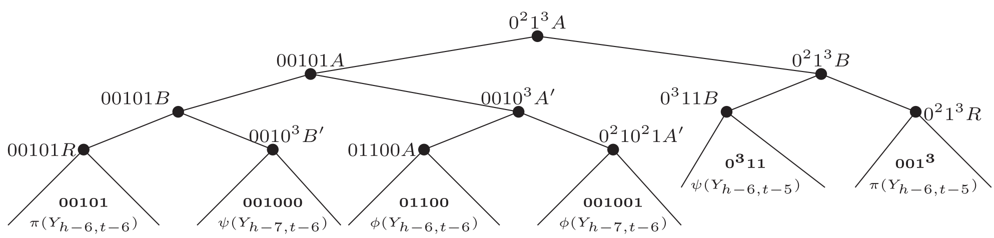


Figure 7. Labelling of $Y^2_{h-3,t-3}$: the embedding of $Y_{h-6,t-6}$ into Q_{p-5}, $Y_{h-7,t-6}$ into Q_{p-6} and $Y_{h-6,t-5}$ into Q_{p-5} follows by induction hypothesis.

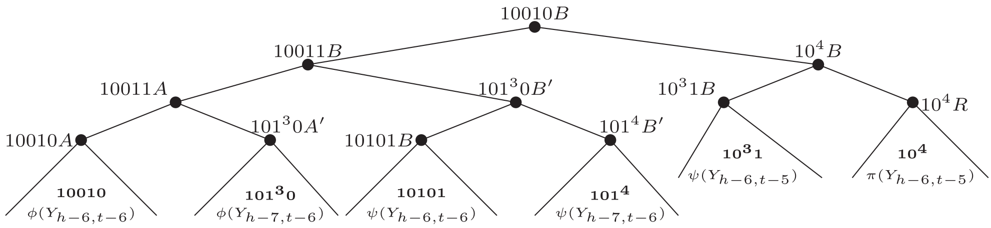


Figure 8. Labelling of $Y^3_{h-3,t-3}$: the embedding of $Y_{h-6,t-6}$ into Q_{p-5}, $Y_{h-7,t-6}$ into Q_{p-6} and $Y_{h-6,t-5}$ into Q_{p-5} follows by induction hypothesis.

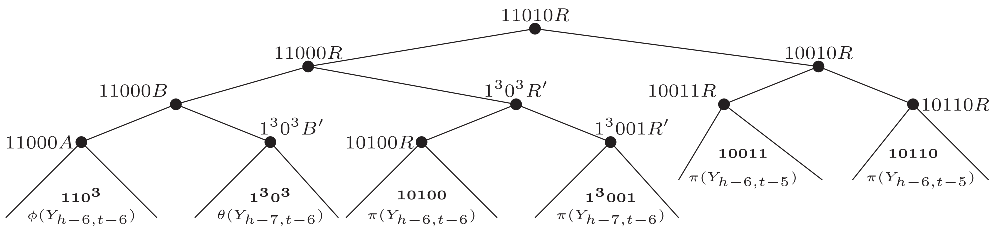


Figure 9. Labelling of $Y^4_{h-3,t-3}$: the embedding of $Y_{h-6,t-6}$ into Q_{m-5}, $Y_{h-7,t-6}$ into Q_{m-6} and $Y_{h-6,t-5}$ into Q_{p-5} follows by induction hypothesis.

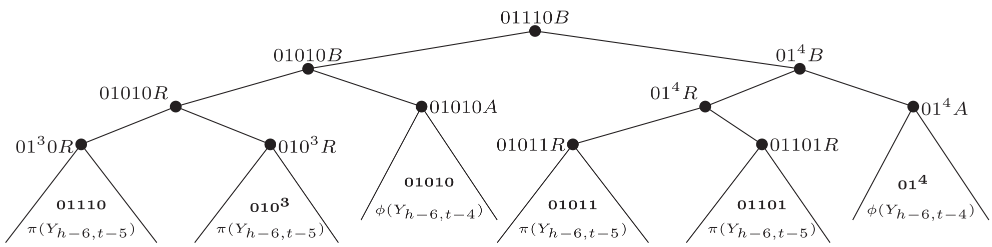


Figure 10. Labelling of $Y^5_{h-3,t-2}$: the embedding of $Y_{h-6,t-5}$ and $Y_{h-6,t-4}$ into Q_{p-5} follows by induction hypothesis.

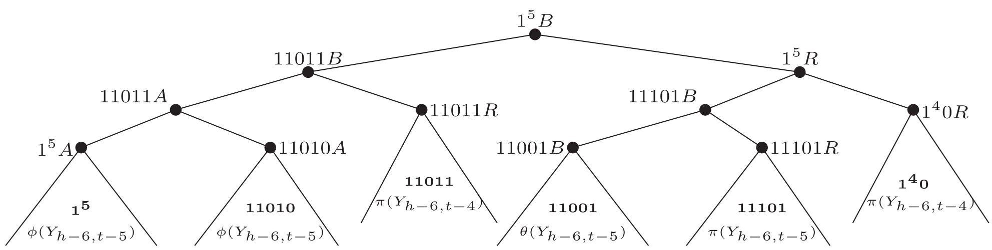


Figure 11. Labelling of $Y^6_{h-3,t-2}$: the embedding of $Y_{h-6,t-5}$ and $Y_{h-6,t-4}$ into Q_{p-5} follows by induction hypothesis.

the length of every label is p. Since Q_p is 2-arc transitive, there exists an automorphism of Q_p which maps the auxiliary path of $Y^*_{h,t}$ onto $(0^{p-2}11, 0^{p-1}1, 0^p)$ thus satisfying the auxiliary condition. Since the labels of the vertices of $Y^*_{h,t}$ are distinct, the provided labelling is an injection and it is a required embedding. ■

4. Conclusion

In this paper, we identified two subclasses of height-balanced trees and proved that every tree of the subclasses is a subtree of the hypercube. The problem of embedding these trees into their optimal hypercube is open. The problem of embedding any height-balanced tree into its optimal hypercube is also open.

Disclosure statement

No potential conflict of interest was reported by the author.

References

[1] G.M. Adelson-Velskii and E.M. Landis, *An algorithm for the organization of information*, Soviet Math. Dokl. 3 (1962), pp. 1259–1262.
[2] S.N. Bhatt and I.C.F. Ipsen, *How to embed trees in hypercubes*, Research Report YALEU/DCS/RR-443, Department of Computer Science, Yale University, 1985.
[3] S.N. Bhatt, F.R.K. Chung, F.T. Leighton, and A.L. Rosenberg, *Optimal Simulations of Tree Machines*, Proceedings of 27th Annual IEEE Symposium on Foundations of Computer Science, Toronto, 1986, pp. 274–282.
[4] S.N. Bhatt, F.R.K. Chung, F.T. Leighton, and A.L. Rosenberg, *Efficient embedding of trees in hypercubes*, SIAM J. Comput. 21(1) (1992), pp. 151–162.
[5] W.K. Chen and M.F.M. Stallmann, *On embedding binary trees into hypercubes*, J. Parallel Distrib. Comput. 24 (1995), pp. 132–138.
[6] S.A. Choudum and I. Raman, *Embedding height balanced trees and Fibonacci trees in hypercubes*, J. Appl. Math. Comput. 30(1–2) (2009), pp. 39–52.
[7] S.A. Choudum and R. Indhumathi, *On embedding subclasses of height balanced trees in hypercubes*, Inform. Sci. 179(9) (2009), pp. 1333–1347.
[8] T. Dvorak, *Dense sets and embedding binary trees into hypercubes*, Discrete Appl. Math. 155 (2007), pp. 506–514.
[9] C.C. Foster, *Information Storage and Retrieval Using AVL Trees*, Proceedings of ACM 20th National Conference, Cleveland, 1965, pp. 192–205.
[10] I. Havel, *On Hamiltonian circuits and spanning trees of hypercubes*, Časopis pro PÄŢstování Matematiky 109(2) (1984), pp. 135–152.
[11] I. Raman and S.A. Choudum, *Embedding certain height-balanced trees and complete p^m-ary trees into hypercubes*, J. Discrete Algorithms 22 (2013), pp. 53–65.
[12] D.E. Knuth, Sorting and Searching. The Art of Computer Programming 2nd ed., Vol. 3, Redwood City, CA, Addison-Wesley Longman Publishing Co., 1998.
[13] S. Lakshmivarahan, J.S. Jwo, and S.K. Dhall, *Symmetry in interconnection networks based on Cayley graphs of permutation groups: A survey*, Parallel Comput. 19(4) (1993), pp. 361–407.
[14] F.T. Leighton, Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes, Morgan Kaufmann, San Mateo, CA, 1992.
[15] M. Medidi and N. Deo, *Parallel dictionaries using AVL-trees*, J. Parallel Distrib. Comput. 49 (1998), pp. 146–155.
[16] B. Monien and H. Sudborough, *Embedding one interconnection network in another*, Comput. Suppl. 7 (1990), pp. 257–282.
[17] A.Y. Wagner, *Embedding arbitrary binary trees in a hypercube*, J. Parallel Distrib. Comput. 7 (1989), pp. 503–520.