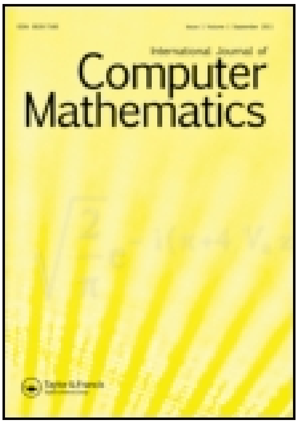

International Journal of Computer Mathematics
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/gcom20

On the *k*-ary hypercube tree and its average distance

M. H. Alsuwaiyel [a]

[a] Department of Information and Computer Science, King Fahd University of Petroleum and Minerals, Dhahran, 31261, Saudi Arabia

Available online: 28 Jun 2011

To cite this article: M. H. Alsuwaiyel (2011): On the *k*-ary hypercube tree and its average distance, International Journal of Computer Mathematics, 88:11, 2258-2274

To link to this article: http://dx.doi.org/10.1080/00207160.2010.544722

International Journal of Computer Mathematics
Vol. 88, No. 11, July 2011, 2258–2274

On the *k*-ary hypercube tree and its average distance

M.H. Alsuwaiyel*

Department of Information and Computer Science, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

(*Received 22 October 2009; revised version received 16 June 2010; second revision received 17 August 2010; third revision received 10 October 2010; accepted 27 November 2010*)

The d-dimensional k-ary hypercube tree $T_k(d)$ is a generalization of the hypercube tree, also known in the literature as the spanning binomial tree. We present some of its structural properties and investigate in detail its average distance. For instance, it is shown that the binary hypercube tree has the anomaly of having two nodes in its centre as opposed to having one in hypercube trees of arity $k > 2$. However, in all dimensions, the centre, centroid and median coincide. We show that its total distance is $\sigma(T_k(d)) = 2\sigma(H_k(d)) - (2/k)\binom{k^d}{2} = dk^{2d} - (d+1)k^{2d-1} + k^{d-1}$, which is minimum. Consequently, for $d \geq 2$, its average distance is $\mu(T_k(d)) = (2d(k-1)k^{d-1})/(k^d - 1) - 2/k$, whose limiting value is 2. This answers a generalization of a conjecture of Dobrynin *et al.* [*Wiener index of trees: Theory and applications*, Acta Appl. Math. 66 (2001), pp. 211–249] for the binary hypercube by the affirmative.

Keywords: hypercube tree; spanning binomial tree; average distance; total distance; Wiener index

2000 AMS Subject Classifications: 05C12; 05C05; 05C85; 90B18; 90B20

1. Introduction

Bhuyan and Agrawal [3] introduced the concept of a generalized hypercube in which k_i nodes are connected along the ith dimension for a total of $\Pi_{i=0}^{d-1} k_i$ nodes. When $k_0 = k_1 \cdots = k_{d-1}$, it is referred to as the k-ary generalized hypercube. A d-dimensional k-ary generalized hypercube, denoted by $H_k(d)$, is a generalization of the binary hypercube $H_2(d)$. It has $n = k^d$ nodes, where each node is uniquely identified with the d-digit base-k number $x_1 x_2 \cdots x_d$, $x_j \in \{0, 1, \ldots, k-1\}$. Two nodes are connected if they differ in exactly one base-k digit. Hence, the degree of each node is $(k-1)d$ and there are a total of $\frac{1}{2}(k-1)dk^d$ edges. $H_k(d)$ has a recursive structure: if $d = 1$, $H_k(1)$ is simply K_k, the complete graph on k nodes, otherwise $H_k(d)$ is constructed from k copies of a $(d-1)$-dimensional hypercube $H_k(d-1)$ by performing the product $K_k \times H_k(d-1)$. Hence,

$$H_k(d) = \underbrace{K_k \times K_k \times \cdots \times K_k}_{d \text{ times}}.$$

*Email: suwaiyel@kfupm.edu.sa

ISSN 0020-7160 print/ISSN 1029-0265 online

DOI: 10.1080/00207160.2010.544722
http://www.informaworld.com

The d-dimensional hypercube spanning tree (or hypercube tree for short) is obtained from the d-dimensional binary hypercube by performing the breadth-first search (BFS) starting from node $00\cdots0$. It is also referred to in the literature as the spanning binomial tree of the hypercube [10,11]. Looking at the connection between the binary hypercube and its spanning tree, it seems natural to generalize the arity of the latter, so that it corresponds to the k-ary hypercube for $k \geq 2$. For this purpose, for $k \geq 2$ and $d \geq 1$, we define the d-dimensional k-ary hypercube tree rooted at node $00\cdots0$ (d zeros), which we will denote by $T_k(d)$, as a rooted tree whose construction is a generalization of that of the hypercube tree or spanning binomial tree. It turns out that the k-ary hypercube tree enjoys some properties which can be thought of as generalizations of those of the binary hypercube tree. One of the important properties that we will investigate in this work is related to its total or average distance. We show that it has the least total (and hence average) distance among all spanning trees of the k-ary hypercube.

Let $G = (V, E)$ be a connected undirected graph on n nodes. For $u, v \in V(G)$, the distance between u and v, denoted by $d_G(u, v)$, is the length of a shortest path between u and v, where the length of a path is defined as the number of edges along the path. For $v \in V(G)$, the distance of v, is defined as

$$d_G(v) = \sum_{u \in V(G)} d_G(v, u).$$

The total distance and average distance of the graph G, denoted, respectively, by $\sigma(G)$ and $\mu(G)$, are defined as

$$\sigma(G) = \frac{1}{2} \sum_{v \in V(G)} d_G(v) \quad \text{and} \quad \mu(G) = \binom{n}{2}^{-1} \sigma(G).$$

The average distance is one of the most important measures of the efficiency of an interconnection network modelled by a graph. It may be a more effective measure of the average performance of a network than its diameter, as it is an indicator of the expected travel time between two randomly chosen points of the network. The total distance has been investigated by several authors (e.g. [13]) and also under different names, such as transmission [15], total routing cost [18], and Wiener index [5,17], with the latter being the oldest and most common. The average distance has also been investigated under the name mean distance [7]. The Wiener index $\sigma(G)$ of a graph G was originally introduced by Wiener [17], and has numerous applications in physical chemistry [9]. It has been extensively studied (see [5] for an excellent survey and results). Applications of the binary hypercube tree in broadcasting and personalized communication and in fault-tolerant computing can be found in [1,6,10,11,14].

In [16], it is shown that the binary hypercube tree is a local optimum with respect to the 1-move heuristic which, starting from a spanning tree T of the hypercube $H_2(d)$, attempts to improve the average distance between pairs of nodes, by adding an edge e of $H_2(d) - T$ and removing an edge e' from the (unique) cycle created by e. In [2], it is shown that the hypercube tree has the minimum total distance among all spanning trees. The work here pertaining to the average distance of the k-ary generalized hypercube tree is essentially an extention/generalization of that in [2]. Given a graph G, let $r(G) = \min\{\sigma(T)/\sigma(G) \mid T \text{ is a spanning tree of } G\}$. Dobrynin *et al.* [5] conjectured that if T is of minimum total distance over all possible spanning trees of $H_2(d)$, then

$$r(H_2(d)) = 2\left(1 - \frac{1}{d}\right) + \frac{1}{d2^{d-1}} \sim 2. \tag{1}$$

Here, we answer by the affirmative a generalization of this conjecture for the case of k-ary generalized hypercube.

2. Preliminaries

For $J = 0, 1, \ldots, k-1$, let $JH_k(d)$ denote the induced subgraph of $H_k(d)$ on node set

$$\{J\, b_1 b_2 \cdots b_{d-1} \mid b_i \in \{0, 1, \ldots, k-1\}\}.$$

That is, $JH_k(d)$, which we will refer to as the J-cube, is the subcube of $H_k(d)$ with all node labels starting with base-k digit J. For instance, the 0-cube is the induced subgraph of $H_k(d)$ on node set $\{0b_1b_2 \cdots b_{d-1} \mid b_i \in \{0, 1, \ldots, k-1\}\}$.

Let G be a connected undirected graph. The eccentricity of a node $v \in V(G)$, denoted by $\mathrm{ecc}(v)$, is the length of a longest of all shortest paths between v and every other node in G. The maximum eccentricity is called the graph diameter. The minimum graph eccentricity is called the graph radius. The centre C of a graph is the set of nodes of graph eccentricity equal to its radius (also called the set of central points). A branch B of a tree T at a node v is a maximal subtree containing v as a leaf. The weight of a branch B, denoted by $\mathrm{bw}(B)$, is the number of edges in B. The branch weight of a node v, denoted by $\mathrm{bw}(v)$, is the maximum branch weight among all branches at v. Equivalently, $\mathrm{bw}(v)$ is the maximum number of nodes in a connected component of $T - v$. The centroid of a tree T is the set of nodes of T with the minimum branch weight. The median $\mathcal{M}$ is defined as the set of nodes with minimum distance. The following theorem is due to Jordan [12].

THEOREM 1 *If $\mathcal{C}$ is the centroid of a tree T of order n, then one of the following holds*: (i) $\mathcal{C} = \{c\}$ *and* $\mathrm{bw}(c) \leq (n-1)/2$, (ii) $\mathcal{C} = \{c_0, c_1\}$ *and* $bw(c_0) = bw(c_1) = n/2$. *In both cases, if* $v \in V(T) - \mathcal{C}$, *then* $\mathrm{bw}(v) \geq n/2$.

Zelinka [19] characterized the set of nodes with minimum distance in a tree.

THEOREM 2 *The set of nodes with minimum distance in a tree T is the centroid $\mathcal{C}$ of T.*

By Theorem 2, the centroid $\mathcal{C}$ and the median $\mathcal{M}$ in a tree are identical [4].

3. The *k*-ary hypercube tree

For $d = 1, 2, \ldots,$ we define the d-dimensional k-ary hypercube tree rooted at node $00\cdots0$ (d zeros), which we denote by $T_k(d)$, as a rooted tree whose set of nodes is $V(T_k(d)) = V(H_k(d))$, and whose set of edges $E(T_k(d))$ is defined by the parent function p as follows. Let $v \in V(T_k(d))$ and $h_k^d(v)$ be the height of v in $T_k(d)$, that is, the maximum number of edges from v to a leaf node in the subtree rooted at v (so the height of the root is the height of the tree, and the height of a leaf node is 0). If v is not the root, then its label has the form

$$xz\underbrace{00\cdots0}_{h_k^d(v)}, \quad \text{where } x \in \{0, 1, \ldots, k-1\}^* \text{ and } z \in \{1, 2, \ldots, k-1\}.$$

So, x is a (possibly empty) sequence of base-k digits and z is a nonzero base-k digit. The parent of v in $T_k(d)$, $d > 0$, is defined as

$$p\left(xz\underbrace{00\cdots0}_{h_k^d(v)}\right) = x\underbrace{00\cdots0}_{h_k^d(v)+1}. \tag{2}$$

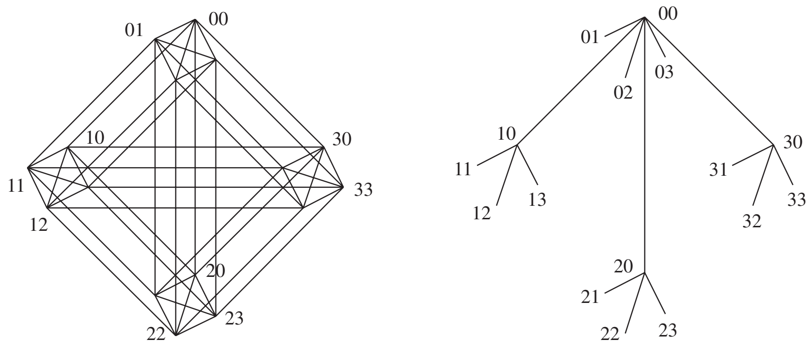

Figure 1. Hypercube $H_4(2)$ and its corresponding rooted tree $T_4(2)$.

If v is the root, then $p(v)$ is undefined. Henceforth, we will refer to the child–parent relationship in Equation (2) as the k-ary hypercube tree property. Figure 1 shows hypercube $H_4(2)$ and its corresponding rooted tree $T_4(2)$.

3.1 *Construction of the k-ary hypercube tree*

The k-ary hypercube tree $T_k(d)$ can be constructed by performing ordinary BFS on $H_k(d)$ starting at node $00\cdots0$. However, using the BFS costs $\Theta(|E(H_k(d))|) = O(|V(H_k(d))| \log |V(H_k(d))|)$ operations, while direct construction using its definition costs only $\Theta(|V(H_k(d))|)$ operations. So, a natural generalization of the known constructions of the spanning binomial tree is more efficient. Here, we generalize and unify two simple construction methods [10,11]. To avoid repetitions, we will set $T_k(1) = K_{1,k-1}$, where $K_{1,k-1}$ is the star graph on k nodes. Assume $d \geq 2$. Then,

(a) $T_k(d)$ is constructed by replacing each node v of $T_k(1)$ with $T_k(d-1)$, and designating the root of the tree replacing the root of $T_k(1)$ as the root of $T_k(d)$.
(b) $T_k(d)$ is constructed from $T_k(d-1)$ by attaching $(k-1)$ leaf nodes to each node in $T_k(d-1)$.

In [16], the property stated in Theorem 3(iii) (Section 4) is used as a construction method for the binary hypercube tree. Given two rooted trees T' and T'', let $T' \odot T''$ denote the tree T obtained by replacing each node in T' by the tree T'', and designating the root of the tree replacing the root of T' as the root of T. Then, construction methods (a) and (b) can be rewritten, respectively, as

$$T_k(d) = T_k(1) \odot T_k(d-1) \tag{3}$$

and

$$T_k(d) = T_k(d-1) \odot T_k(1). \tag{4}$$

LEMMA 1 *The operation $\odot$ is associative.*

Proof For $i, j, l \geq 1$, let $T_k(i)$, $T_k(j)$ and $T_k(l)$ be k-ary hypercube tree s of dimensions i, j and l, respectively. By definition of the operation $\odot$,

$$(T_k(i) \odot T_k(j)) \odot T_k(l) = T_k(i+j) \odot T_k(l) = T_k(i+j+l),$$

and likewise,

$$T_k(i) \odot (T_k(j) \odot T_k(l)) = T_k(i) \odot T_k(j+l) = T_k(i+j+l).$$

Thus, $\odot$ is associative. ■

By Lemma 1, we have the following general definition of $T_k(d)$. For $d \geq 2$ and $1 \leq j \leq d-1$,

$$T_k(d) = T_k(j) \odot T_k(d-j). \tag{5}$$

Consequently, for any $d > 1$,

$$T_k(d) = \underbrace{T_k(1) \odot T_k(1) \odot \cdots T_k(1)}_{d \text{ times}}.$$

The set of nodes of $T_k(d)$ is

$$V(T_k(d)) = \{uv | u = x_1, x_2, \ldots, x_j \in T_k(j), v = y_1 y_2 \cdots y_{d-j} \in T_k(d-j)\}, \tag{6}$$

and its set of edges is

$$E(T_k(d)) = E_j \cup E_{d-j}, \tag{7}$$

where

$$E_j = \{(u0^{d-j}, v0^{d-j}) | (u, v) \in E(T_k(j))\}, \tag{8}$$

and

$$E_{d-j} = \{(wx, wy) | w \in V(T_k(j)) \text{ and } (x, y) \in E(T_k(d-j))\}. \tag{9}$$

In other words, the set of edges in $T_k(d)$ are the union of two sets: (1) E_j is the set of edges in $T_k(j)$ with their end nodes appended by $d-j$ zeros, and (2) E_{d-j} consists of k^j copies of the set of edges in $T_k(d-j)$, where the end nodes of the edges in each copy are prefixed by the label of a node in $T_k(j)$. So, E_{d-j} is the set of edges resulting from replacing each node of $T_k(j)$ with one copy of $T_k(d-j)$.

LEMMA 2 *The graph obtained using Equation* (5) *is the d-dimensional k-ary hypercube tree.*

Proof Let $T = T' \odot T''$, where T' and T'' are two k-ary hypercube tree s of dimensions j and $d-j$, respectively, with $1 \leq j < d$. By Equation (6),

$$|V(T_k(d))| = k^j \times k^{d-j} = k^d,$$

and by Equations (7)–(9),

$$|E(T)| = |E(T')| + k^j |E(T'')| = (k^j - 1) + k^j (k^{d-j} - 1) = k^d - 1.$$

Since T is connected, it follows that it is a spanning tree for $H_k(d)$. So, it only remains to show that T has the k-ary hypercube tree property. To this end, assume inductively that the k-ary hypercube tree property is satisfied by both T' and T''. Let (u, v) be an edge of T, where u is the parent of v. By Equation (6), $u = w_1 x$ and $v = w_2 y$, where $w_1, w_2 \in V(T')$ and $x, y \in V(T'')$. If $x = y = 0^{d-j}$, then (w_1, w_2) is an edge of T'. On the other hand, if $y \neq 0^{d-j}$, then (x, y) is an edge of T''. By induction, in both cases, the edge (u, v) in T satisfies Equation (2). Since (u, v) is arbitrary, it follows that the k-ary hypercube tree property is satisfied in T. This, by definition, implies that T is a d-dimensional k-ary hypercube tree. ■

In the rest of the paper, we will exclusively make use of constructions (a) and (b). Also, if $d = 0$, both $H_k(0)$ and $T_k(0)$ consist of one node and no edges. For $J = 0, 1, \ldots, k-1$, let $JT_k(d)$ denote the induced subgraph of $T_k(d)$ on node set

$$\{J\, b_1 b_2 \cdots b_{d-1} \mid b_i \in \{0, 1, \ldots, k-1\}\}.$$

That is, $JT_k(d)$, which we will refer to as the J-tree, is the subtree of $T_k(d)$ with all node labels starting with base-k digit J. For instance, the 0-tree is the induced subtree of $T_k(d)$ on node set $\{0 b_1 b_2 \cdots b_{d-1} \mid b_i \in \{0, 1, \ldots, k-1\}\}$.

4. Some properties of the *k*-ary hypercube tree

The following theorem summarizes some of the properties of the k-ary hypercube tree. For brevity, let $\zeta(v) = d$, if v is the root; otherwise $\zeta(v)$ is the number of zeros to the right of the rightmost nonzero digit in the k-ary representation of node v. So, $\zeta(v)$ is simply the height of node v.

THEOREM 3 *Let $T_k(d)$ be a d-dimensional k-ary hypercube tree of dimension $d > 0$. Then,*

(i) *the number of nodes in the subtree rooted at node v is $k^{\zeta(v)}$;*
(ii) *for $0 \le j \le d$, the number of nodes at level j is $(k-1)^j \binom{d}{j}$;*
(iii) *$T_k(d)$ consists of the following subtrees each connected to the root by an edge: $(k-1)$ subtrees of dimension $d-1$, $(k-1)$ subtrees of dimension $d-2, \ldots, (k-1)$ subtrees of dimension 0;*
(iv) *the centroid of $T_k(d)$ is $\mathcal{C} = \{0^d, 10^{d-1}\}$ if $k = 2$, and $\mathcal{C} = \{0^d\}$ if $k > 2$;*
(v) *$C = \mathcal{C} = \mathcal{M}$, that is, the centre, the centroid and the median are identical in $T_k(d)$;*
(vi) *for all $v \in V(T_k(d)) d_{T_k(d)}(0^d, v) = d_{H_k(d)}(0^d, v)$;*
(vii) *the radius of $T_k(d)$ is $\text{rad}(T_k(d)) = d$, and the diameter of $T_k(d)$ is $D(T_k(d)) = 2d - 1$ if $k = 2$ and $D(T_k(d)) = 2d$ if $k > 2$.*

Proof (i) Let $\zeta(v) = j$. If $j \in \{0, d\}$, then the result is obviously true. So, assume that $0 < j < d$, and let T be the subtree rooted at node v. By Equation (5), $T_k(d) = T_k(d-j) \odot T_k(j)$. Clearly, T is isomorphic to $T_k(j)$, and hence contains k^j nodes.

(ii) Assume the result is true for $d-1$. Since $T_k(d)$ is constructed from $T_k(d-1)$ by attaching $(k-1)$ leaf nodes to each node in $T_k(d-1)$, the number of nodes at level $j > 0$ in $T_k(d)$ is equal to the number of nodes at level j in $T_k(d-1)$ plus $(k-1)$ times the number of nodes at level $j-1$ in $T_k(d-1)$. By induction, their total is

$$(k-1)^j \binom{d-1}{j} + (k-1)\left((k-1)^{j-1}\binom{d-1}{j-1}\right) = (k-1)^j \binom{d}{j}.$$

(iii) By construction, there are $(k-1)$ trees of dimension $(d-1)$ linked to the root by an edge plus one $(d-1)$-dimensional tree T' whose root is the root of $T_k(d)$. So, expanding T', there are $(k-1)$ trees of dimension $(d-2)$ connected to the root by an edge plus one $(d-2)$-dimensional tree T'' whose root is the root of T' (which is the root of $T_k(d)$), and so on. It follows that there are $(k-1)$ subtrees of dimension $d-1$, $(k-1)$ subtrees of dimension $d-2$, …, $(k-1)$ subtrees of dimension 0, each linked to the root by an edge (Figure 2).

(iv) Let $v \in V(T_k(d))$. If v is the root, then removing v leaves at least one subtree of size k^{d-1}, which is maximum. So assume v is not the root. By Part (i), the number of nodes in the subtree rooted at node v is $k^{\zeta(v)}$. Hence, removing v leaves one connected component of

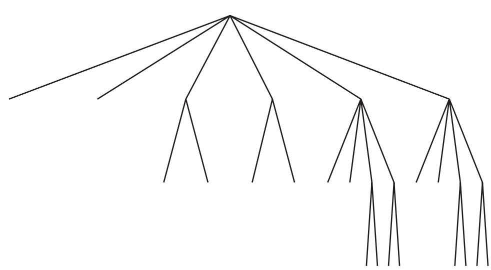

Figure 2. $T_3(3)$ consists of two $T_3(2)$'s, two $T_3(1)$'s and two $T_3(0)$'s linked to its root.

Table 1. Distribution of nodes to levels in $T_4(d)$ for $0 \leq d \leq 4$.

d	0	1	2	3	4
0	1				
1	1	3			
2	1	6	9		
3	1	9	27	27	
4	1	12	54	108	81

size $k^d - k^{\zeta(v)}$. Consequently, if $k = 2$, then letting $v = 10^{d-1}$, we have that $\text{bw}(10^{d-1}) = 2^{d-1} = \text{bw}(0^d)$, and if $k > 2$, then,

$$\text{bw}(v) = k^d - k^{\zeta(v)} \geq k^d - k^{d-1} = (k-1)k^{d-1} > k^{d-1} = \text{bw}(0^d).$$

By Theorem 1, it follows that the centroid is $\mathcal{C} = \{0^d, 10^{d-1}\}$ if $k = 2$, and $\mathcal{C} = \{0^d\}$ if $k > 2$.

(v) Let $T = T_k(d)$. Starting with T, remove all leaf nodes to obtain T'. Next, remove all leaf nodes of T' and so on until the size of the tree is reduced to k. This is the reverse of the construction method (b) whose base case implies that the resulting tree must be either K_2 if $k = 2$ or $K_{1,k-1}$ if $k > 2$. If $k > 2$, removing all nodes in $K_{1,k-1}$ of degree 1 leaves exactly one node. Thus, if $k = 2$, the remaining nodes are 0^d and 10^{d-1}, and if $k > 2$, the remaining node is 0^d. By Part (iv), the remaining node(s) constitute the centroid of T. But this is the same method used for finding the centre of a tree [4]. This implies that this procedure of successive elimination of leaf nodes when applied on $T_k(d)$ finds both its centre and centroid. Finally, since in any tree $\mathcal{C} = \mathcal{M}$ (Theorem 2), we have that $C = \mathcal{C} = \mathcal{M}$.

(vi) This follows from the fact that $T_k(d)$ can be obtained by performing the BFS on $H_k(d)$ starting from node $00\cdots 0$.

(vii) By part (vi), $\text{rad}(T_k(d)) = \text{rad}(H_k(d)) = \text{d}$. Simple induction on d shows that the diameter of $T_k(d)$ is $D(T_k(d)) = 2d - 1$ if $k = 2$, and $D(T_k(d)) = 2d$ if $k > 2$. In fact, this follows from a general result on trees that can be found in [4]: in a tree T, $D(T) = 2\,\text{rad(T)}$ if $|\mathcal{C}| = 1$, and $D(T) = 2\,\text{rad(T)} - 1$ if $|\mathcal{C}| = 2$. ■

By Theorem 3(ii), $(1 + (k-1)x)^d$ is the generating function for the number of nodes at level d in $T_k(d)$. The distribution of nodes to levels in $T_4(d)$ for $0 \leq d \leq 4$ is shown in Table 1. For instance, in $T_4(2)$, which is shown in Figure 1, levels 0, 1 and 2 have, respectively, 1, 6 and 9 nodes.

5. Total distance of the *k*-ary hypercube tree

First, we compute the distance of the root of $T_k(d)$,

$$d_{T_k(d)}(0^d) = \sum_{v \in V(T_k(d))} d_{T_k(d)}(0^d, v),$$

and establish some relationships between distances in the hypercube tree $T_k(d)$ and its corresponding hypercube graph $H_k(d)$.

LEMMA 3 *Let $T_k(d)$ be a d-dimensional k-ary hypercube tree. Then,*

(i) $d_{H_k(d)}(0^d) = d_{T_k(d)}(0^d) = d(k-1)k^{d-1}$.
(ii) $\sigma(H_k(d)) = \frac{1}{2}d(k-1)k^{2d-1}$.

Proof (i) By Theorem 3(ii), the number of nodes at distance j from the root, $0 \le j \le d$, is equal to $(k-1)^j \binom{d}{j}$. Hence, the distance of the root $d_{T_k(d)}(0^d)$ is computed as

$$\begin{aligned} d_{T_k(d)}(0^d) &= \sum_{j=1}^{d} j(k-1)^j \binom{d}{j} \\ &= d(k-1)\sum_{j=1}^{d}(k-1)^{j-1}\binom{d-1}{j-1} \\ &= d(k-1)\sum_{j=0}^{d-1}(k-1)^{j}\binom{d-1}{j} \\ &= d(k-1)(1+(k-1))^{d-1} \\ &= d(k-1)k^{d-1}. \end{aligned}$$

Since, by Theorem 3(vi), for all $v \in V(T_k(d)) d_{T_k(d)}(0^d, v) = d_{H_k(d)}(0^d, v)$, it follows that $d_{H_k(d)}(0^d) = d_{T_k(d)}(0^d) = d(k-1)k^{d-1}$.

(ii) By symmetry of the k-ary hypercube, the distance of each node is equal to $d_{H_k(d)}(0^d)$. Hence, by part (i), the total distance of the d-dimensional k-ary hypercube $H_k(d)$ is

$$\sigma(H_k(d)) = \frac{k^d}{2} \times d_{H_k(d)}(0^d) = \frac{1}{2} d(k-1)k^{2d-1}. \qquad \blacksquare$$

In the remainder of this section, let $\kappa = (2(k-1))/k$.

LEMMA 4 *For all $k \ge 2$, $d \ge 2$, the total distance of the d-dimensional k-ary hypercube tree $T_k(d)$ can be expressed as*

$$\sigma(T_k(d)) = k\sigma(T_k(d-1)) + \kappa\sigma H_k(d).$$

Proof Let u and v be two nodes in subtrees $IT_k(d)$ and $JT_k(d)$, respectively. Let $l_{i,j}$, $i < j$, be the length of the path between the root of $IT_k(d)$ and the root of $JT_k(d)$. That is, $l_{i,j}$ is the length of the path between nodes $I0^{d-1}$ and $J0^{d-1}$. Then,

$$l_{i,j} = \begin{cases} 1 & \text{if } i = 0, \\ 2 & \text{if } i \ne 0. \end{cases}$$

The distance between u and v is (Figure 3)

$$d(u, v) = d(u, I0^{d-1}) + l_{i,j} + d(J0^{d-1}, v).$$

Summing over all nodes $u \in IT_k(d)$ and $v \in JT_k(d)$, we have

$$\begin{aligned} s_{i,j} &= \sum_{u \in IT_k(d)} \sum_{v \in JT_k(d)} d(u, v) \qquad (10) \\ &= \sum_{u \in IT_k(d)} \sum_{v \in JT_k(d)} (d(u, I0^{d-1}) + l_{i,j} + d(J0^{d-1}, v)) \\ &= k^{d-1}\left(\sum_{u \in IT_k(d)} d(u, I0^{d-1})\right) + (k^{d-1}k^{d-1}l_{i,j}) + \left(k^{d-1}\sum_{v \in JT_k(d)} d(J0^{d-1}, v)\right) \end{aligned}$$

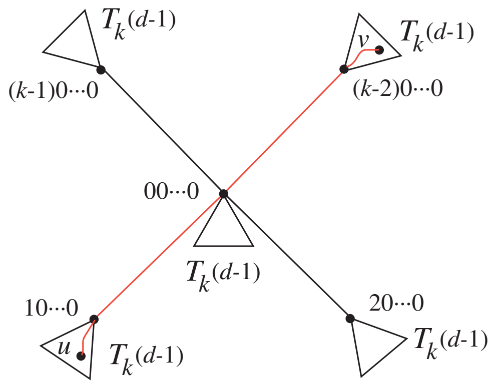

Figure 3. Proof of Lemma 4. The connection edges form $T_k(1)$.

$$= 2k^{d-1}\left(\sum_{u\in IT_k(d)} d(u, I0^{d-1})\right) + k^{2d-2}l_{i,j} \qquad \text{(by symmetry)}$$
$$= 2k^{d-1}d_{T_k(d-1)}(0^{d-1}) + k^{2d-2}l_{i,j}. \tag{11}$$

Let s denote the sum of all distances between all pairs of nodes u and v, where u and v belong to different $(d-1)$-dimensional subtrees. Then, summing over all pairs of subtrees and using Equation (11), we have

$$s = \sum_{i<j} s_{i,j} = \binom{k}{2}(2k^{d-1}d_{T_k(d-1)}(0^{d-1})) + k^{2d-2}\sum_{i<j} l_{i,j}.$$

By definition of σ, $\sum_{i<j} l_{i,j} = \sigma(T_k(1)) = \sigma K_{1,k-1} = (k-1)^2$. Hence,

$$\begin{aligned} s &= 2\binom{k}{2}k^{d-1}d_{T_k(d-1)}(0^{d-1}) + k^{2d-2}\sigma(T_k(1)) \\ &= k(k-1)k^{d-1}((d-1)(k-1)k^{d-2}) + k^{2d-2}(k-1)^2 \quad \text{(by Lemma 3(i))} \\ &= (d-1)(k-1)^2k^{2d-2} + k^{2d-2}(k-1)^2 \\ &= dk^{2d-2}(k-1)^2 \\ &= \frac{1}{2}d\kappa(k-1)k^{2d-1} \\ &= \kappa\sigma(H_k(d)) \quad \text{(by Lemma 3(ii))}. \end{aligned}$$

Since the total distance $\sigma(T_k(d))$ is the sum of distances within the k subtrees plus the total interdistances between nodes in different subtrees, we conclude that $\sigma(T_k(d))$ can be expressed by the recurrence

$$\sigma(T_k(d)) = \begin{cases} (k-1)^2 & \text{if } d = 1, \\ k\sigma(T_k(d-1)) + \kappa\sigma(H_k(d)) & \text{if } d > 1. \end{cases} \qquad \blacksquare$$

Let $T = (V, E)$ be a tree and $e = (u, v)$ be an edge of T. Let $n_u(e)$ denote the number of nodes of T lying closer to u than v, and let $n_v(e)$ denote the number of nodes of T lying closer to v than u. The following theorem was discovered by Wiener [17].

THEOREM 4 *Let $T = (V, E)$ be a tree. Then, $\sigma(T) = \sum_{e \in E(T)} n_u(e) n_v(e)$.*

LEMMA 5 *For all $k \geq 2$, $d \geq 2$, the total distance of the d-dimensional k-ary hypercube tree $T_k(d)$ can be expressed as*

$$\sigma(T_k(d)) = k^2\, \sigma(T_k(d-1)) + \kappa \binom{k^d}{2}.$$

Proof By Theorem 4,

$$\sigma(T_k(d-1)) = \sum_{(u,v) \in T_k(d-1)} n_u n_v. \tag{12}$$

Since $T_k(d)$ is constructed from $T_k(d-1)$ by replacing each node in $T_k(d-1)$ with $K_{1,k-1}$, both n_u and n_v in Equation (12) get multiplied by a factor of k, and $T_k(d)$ will have $(k-1)k^{d-1}$ leaf nodes whose incident edges' contribution to $\sigma(T_k(d))$ is $1 \times (k^d - 1)$ each. Hence,

$$\begin{aligned}\sigma(T_k(d)) &= k \times k \times \sigma(T_k(d-1)) + ((k-1)k^{d-1})(k^d - 1) \\ &= k^2 \sigma(T_k(d-1)) + \kappa \binom{k^d}{2}.\end{aligned}$$

■

THEOREM 5 *The total distance of the d-dimensional k-ary hypercube tree $T_k(d)$ is*

$$\sigma(T_k(d)) = 2\sigma(H_k(d)) - \frac{2}{k}\binom{k^d}{2} = dk^{2d} - (d+1)k^{2d-1} + k^{d-1}.$$

Proof By Lemma 4,

$$k\sigma(T_k(d)) = k^2 \sigma(T_k(d-1)) + k\kappa\sigma(H_k(d)).$$

By Lemma 5,

$$\sigma(T_k(d)) = k^2\, \sigma(T_k(d-1)) + \kappa \binom{k^d}{2}.$$

Combining and simplifying yields

$$\begin{aligned}\sigma(T_k(d)) &= 2\, \sigma(H_k(d)) - \frac{2}{k}\binom{k^d}{2} \\ &= dk^{2d} - (d+1)k^{2d-1} + k^{d-1}.\end{aligned}$$

■

Table 2 lists the values of $\sigma(T_k(d))$ for four k-ary hypercube tree s of order 256. As expected, the larger the value of k, the lesser the total distance.

Table 2. $\sigma(T_k(d))$ for trees of order 256.

k	d	$\sigma(T_k(d))$
2	8	229,504
4	4	180,288
16	2	118,800
256	1	65,025

6. Optimality of the k-ary hypercube tree

This section is concerned with the optimality of the k-ary hypercube tree $T_k(d)$. It will be shown that it has the least total distance among all spanning trees of the k-ary hypercube $H_k(d)$. The basic idea is to show that the spanning tree must contain all edges of the form $(i0^{d-1}, 0^d)$, $0 < i < k$, and that the tree does not contain edges of the form (u, v), where u and v are in different subcubes.

In this section, we will use the letter G to denote an arbitrary spanning tree for the k-ary hypercube. For convenience and conformity, we will also use small letters to identify subcubes (e.g. $jH_k(d)$ for $JH_k(d)$). Let $G_k(d)$ be a spanning tree for $H_k(d)$ rooted at 0^d in its centroid. For $1 \le j \le k-1$, if $e_j = (j0^{d-1}, 0^d) \in E(G_k(d))$, then let S_j denote the set of nodes whose path to the root 0^d of $G_k(d)$ contains the edge e_j; otherwise S_j is empty. If S_j is nonempty, then we define the subtree rooted at $j0^{d-1}$ to be the tree induced by the nodes in S_j. If S_j is empty, then there is no subtree rooted at $j0^{d-1}$. The set of remaining nodes, which is equal to

$$S_0 = V(G_k(d)) \setminus \bigcup_{1 \le j < k} S_j,$$

is the set of nodes of the subtree of $G_k(d)$ that has as its root 0^d, the root of $G_k(d)$. Note that S_0 cannot be empty, as it contains 0^d. The tree shown in Figure 4(a) contains four subtrees, while the one in Figure 4(b) contains only two subtrees rooted at 0^d and 20^{d-1}. Call a tree $G_k(d)$ 'good' if, in addition to the subtree rooted at 0^d, it also has $k-1$ subtrees rooted at $10^{d-1}, \ldots, (k-1)0^{d-1}$; otherwise it is 'not good'. The tree shown in Figure 4(a) is good, while the one in Figure 4(b) is not.

LEMMA 6 *If $G_k(d)$ is not good, then there is a spanning tree $\hat{G}_k(d)$ that is good with the property that $\sigma(\hat{G}_k(d)) \le \sigma(G_k(d))$.*

Proof We will treat the case in which $G_k(d)$ has $k-1$ subtrees, as the argument is similar if the number of subtrees is less than $k-1$. Note that there is at least one subtree, since, by definition of 'good tree', the subtree rooted at 0^d is part of $G_k(d)$. Suppose that for some j, $0 < j < k$, $j0^{d-1}$ is not connected to the root 0^d of $G_k(d)$. For some $i \ne j$, let $i0^{d-1}$ be the node that is closest in the tree $G_k(d)$ to $j0^{d-1}$ among all nodes of the form $x0^{d-1}$, $0 \le x < k$. Let $i0^{d-1} = v_0, v_1, \ldots, v_l, v_{l+1}, \ldots, v_m = j0^{d-1}$, for some $m > l \ge 0$, be the (unique) path from $i0^{d-1}$ to $j0^{d-1}$ in $G_k(d)$ with the property that l is the largest index such that $v_l \notin V(jH_k(d))$ and $v_{l+1} \in V(jH_k(d))$. Let S_j be the set of nodes in $V(G_k^i(d-1))$ whose path to the root of $G_k(d)$ contains the edge (v_l, v_{l+1}), and $S_i = V(G_k^i(d-1)) \setminus S_j$. Let G_k^i and G_k^j be the two subgraphs induced by the nodes in S_i and S_j, respectively (Figure 5).

Let $e = (v_l, v_{l+1})$ and $e' = (v_0, v_m) = (i0^{d-1}, j0^{d-1})$. Let $\bar{G}_k(d)$ be the tree obtained from $G_k(d)$ by deleting the edge e and adding the edge e'. Define the corresponding subtree $\bar{G}_k^i(d-1)$ of

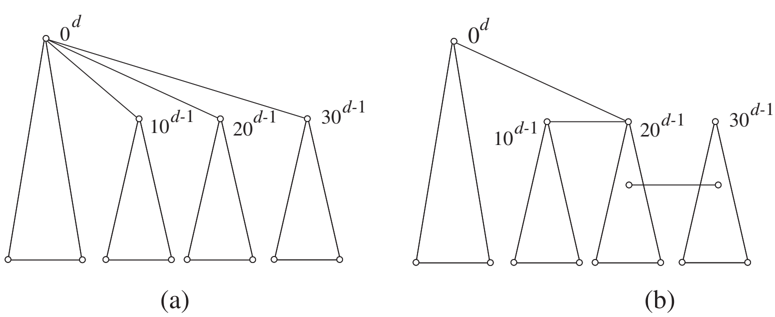

Figure 4. (a) Good tree with four subtrees. (b) A tree that is not good with two subtrees.

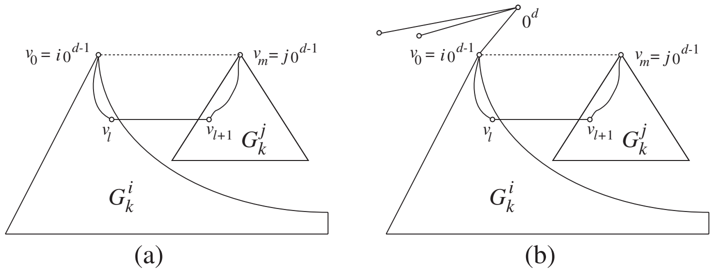

Figure 5. Subtree $G_k^i(d-1)$. (a) $i = 0$. (b) $i > 0$; the subtree rooted at 0^d is not shown.

$\bar{G}_k(d)$ by $\bar{G}_k^i(d-1) = G_k^i(d-1) - e + e'$. Let $n_i = |V(G_k^i)|$ and $n_j = |V(G_k^j)|$. We now show that this replacement of edges will not increase the total distance of $G_k(d)$, that is, $\sigma(\bar{G}_k(d)) \leq \sigma(G_k(d))$. For brevity, let d_l and d_{l+1} denote the distances of v_l and v_{l+1} in G_k^i and G_k^j with respect to $G_k^i(d-1)$ (before the replacement of edges). Similarly, let d_0 and d_m be the distances of v_0 and v_m in G_k^i and G_k^j with respect to $\bar{G}_k^i(d-1)$ (after the replacement of edges). Then, we have

$$\sigma(G_k^i(d-1)) = n_i d_{l+1} + n_j d_l + n_i n_j + \sigma(G_k^i) + \sigma(G_k^j)$$

and

$$\sigma(\bar{G}_k^i(d-1)) = n_i d_m + n_j d_0 + n_i n_j + \sigma(G_k^i) + \sigma(G_k^j).$$

Since $d_0 \leq d_l$ and $d_m \leq d_{l+1}$, it follows that $\sigma(\bar{G}_k^i(d-1)) \leq \sigma(G_k^i(d-1))$. It is fairly easy to see that the sum of the remaining distances in $\bar{G}_k(d)$, if any, will not increase after the edge replacement. Consequently, $\sigma(G_k(d))$ will not increase as a result of the edge replacement.

Now, if $i = 0$, then $\bar{G}_k(d)$ is good, and therefore set $\hat{G}_k(d) = \bar{G}_k(d)$ (Figure 5(a)). Hence, in the rest of the proof, we will assume that $i > 0$ (Figure 5(b)). We will also be primarily dealing with $\bar{G}_k(d)$ instead of $G_k(d)$. By Theorem 1, $\text{bw}(0^d) \leq n/2$, and since $i0^{d-1}$ and $j0^{d-1}$ belong to the same component of $\bar{G}_k(d) - 0^d$, we must have

$$n_i + n_j \leq \frac{n}{2}, \tag{13}$$

which follows from the definition of $\text{bw}(0^d)$ as the maximum number of nodes in a connected component of $\bar{G}_k(d) - 0^d$. Construct $\hat{G}_k(d)$ from $\bar{G}_k(d)$ by deleting the edge $(i0^{d-1}, j0^{d-1})$ and adding the edge $(j0^{d-1}, 0^d)$. Let $\hat{G}_k^i(d-1)$ and $\hat{G}_k^j(d-1)$ be the two subtrees of $\hat{G}_k(d)$ rooted at $i0^{d-1}$ and $j0^{d-1}$, respectively. Now, we show that the total distance of $\hat{G}_k(d)$ is at most that of $\bar{G}_k(d)$. The distance from any node in $\bar{G}_k^i(d-1)$ to any node in $\bar{G}_k^i(d-1)$ will be increased by 1, and the distance from any node in $\bar{G}_k^i(d-1)$ to the rest of the nodes in $\bar{G}_k(d)$ will be decreased by 1. Hence, the net change in the total distance is

$$\begin{aligned}\sigma(\hat{G}_k(d)) - \sigma(\bar{G}_k(d)) &= n_j \times n_i - n_j \times (n - n_i - n_j) \\ &= n_j(2n_i + n_j - n).\end{aligned}$$

Suppose that $\sigma(\hat{G}_k(d)) > \sigma(\bar{G}_k(d))$. Then, $2n_i + n_j - n > 0$, or

$$n_i > \frac{n - n_j}{2}. \tag{14}$$

Combining Equations (13) and (14) yields

$$n_j < 0,$$

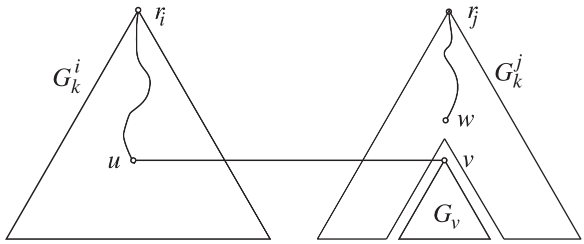

Figure 6. A cross edge (u, v).

contradicting the fact that $n_j > 0$. It follows that $\sigma(\hat{G}_k(d)) \leq \sigma(\bar{G}_k(d))$. Since $\sigma(\bar{G}_k(d)) \leq \sigma(G_k(d))$ as shown above, we have that $\sigma(\hat{G}_k(d)) \leq \sigma(G_k(d))$. ■

Recall that a subtree $G_k^j(d-1)$, $j > 0$, has as its set of nodes those whose path to the root of $G_k(d)$ contains the edge $(j0^{d-1}, 0^d)$, and the remaining nodes constitute the set of nodes of the subtree $G_k^0(d-1)$. If v is a node in subtree $G_k^i(d-1)$, $j \geq 0$, then the set of nodes consisting of v and its descendants (those whose path to the root contains v) will also be called a subtree. For instance, in Figure 6, G_v is a subtree rooted at v. Let $G_k(d)$ be a spanning tree for $H_k(d)$ and $G_k^i(d-1)$ and $G_k^i(d-1)$ two subtrees of $G_k(d)$. We will call an edge $e = (u, v)$ a 'cross edge' if $e \in E(G_k^i(d-1))$, $u \in V(iH_k(d))$ and $v \in V(jH_k(d))$. The other case in which $e \in E(G_k^i(d-1))$ is symmetrical. Suppose that $u \in V(G_k^i(d-1)) \cap V(iH_k(d))$, $v \in V(G_k^i(d-1)) \cap V(jH_k(d))$ and $w \in V(G_k^j(d-1)) \cap V(jH_k(d))$. Let G_v be the subtree rooted at v, and assume that $V(G_v) \subseteq V(jH_k(d))$. We will also assume without loss of generality that (w, v), which is not in $E(G_k^j(d-1))$, is the last edge on the path from r_j to v, which is entirely contained in $jH_k(d)$. It is also reasonable to assume that the two paths from r_i to u and from r_j to w are of shortest length in $H_k(d)$. Since u belongs to $V(iH_k(d))$, v belongs to $V(jH_k(d))$ and there is an edge connecting u and v, the nodes u and v must have the form $u = ix_1 \cdots x_{d-1}$ and $v = jx_1 \cdots x_{d-1}$ for some $x_l \in \{0, 1, \ldots, k-1\}$, $l = 1, \ldots, d-1$. From this fact as well as from the symmetry present in $H_k(d)$, and the above assumption, it can be concluded that $d(r_i, u) = d(r_j, w)$. Now, since the node w is on the shortest path from r_j to v, it follows that

$$d(r_j, w) < d(r_i, u). \tag{15}$$

LEMMA 7 *Let $\hat{G}_k(d)$ be the tree constructed from $G_k(d)$ by deleting the edge (u, v) and adding the edge (w, v). Then, $d_{\hat{G}_k^i(d-1)}(r_i) + d_{\hat{G}_k^j(d-1)}(r_j) \leq d_{G_k^i(d-1)}(r_i) + d_{G_k^j(d-1)}(r_j)$.*

Proof (Figure 6). Assume for simplicity that there is only one cross edge, namely (u, v), as the generalization to more than one cross edge can be managed by removing one cross edge at a time starting, say, at the one closest to the root of $G_k(d)$. Let $|V(G_v)| = m$. Let $d_{G_v}(v)$ be the distance of v in G_v, which is equal to $\sum_{z \in G_v} d(v, z)$. Since there are m paths from r_i to the nodes in G_v,

$$\begin{aligned} d_{\hat{G}_k^i(d-1)}(r_i) &= d_{G_k^i(d-1)}(r_i) - md(r_i, u) - md(u, v) - d_{G_v}(v) \\ &= d_{G_k^i(d-1)}(r_i) - md(r_i, u) - m - d_{G_v}(v), \end{aligned} \tag{16}$$

and

$$\begin{aligned} d_{\hat{G}_k^j(d-1)}(r_j) &= d_{G_k^j(d-1)}(r_j) + md(r_j, w) + md(w, v) + d_{G_v}(v) \\ &= d_{G_k^j(d-1)}(r_j) + md(r_j, w) + m + d_{G_v}(v) \\ &\leq d_{G_k^j(d-1)}(r_j) + m\, d(r_i, u) + m + d_{G_v}(v), \end{aligned} \tag{17}$$

where the last inequality follows from Equation (15). Combining Equations (16) and (17) yields

$$d_{\hat{G}_k^i(d-1)}(r_i) + d_{\hat{G}_k^j(d-1)}(r_j) \leq d_{G_k^i(d-1)}(r_i) + d_{G_k^j(d-1)}(r_j). \tag{18}$$

■

Applying Lemma 7 on each pair of subtrees $G_k^i(d-1)$ and $G_k^j(d-1)$ results in k (possibly new) spanning trees $\hat{G}_k^l(d-1)$ for $lH_k(d)$, $0 \leq l < k$, with the property that no two of them share a cross edge. Now, applying Equation (18) on all pairs of subtrees, we have

$$\sum_{0 \leq i < j < k} (d_{G_k^i(d-1)}(r_i) + d_{G_k^j(d-1)}(r_j)) \geq \sum_{0 \leq i < j < k} (d_{\hat{G}_k^i(d-1)}(r_i) + d_{\hat{G}_k^j(d-1)}(r_j)).$$

Since for $0 \leq l < k$, $\hat{G}_k^l(d-1)$ is a spanning tree for $lH_k(d)$, we have as a consequence

$$\sum_{0 \leq i < j < k} (d_{G_k^i(d-1)}(i0^{d-1}) + d_{G_k^j(d-1)}(j0^{d-1})) \geq \sum_{0 \leq i < j < k} (d_{iH_k(d)}(i0^{d-1}) + d_{jH_k(d)}(j0^{d-1})). \tag{19}$$

Note that applying Lemma 7 on $G_k(d)$ until *all* cross edges are removed will not increase its total distance, that is, $\sigma(\hat{G}_k(d)) \leq \sigma(G_k(d))$. We justify this in connection with the example graph shown in Figure 6. Since only those distances involving nodes in $V(G_v)$ will be affected by the replacement of edges, and since the sum of the distances of the two roots will not increase, it must be the case that the sum of all distances involving those nodes in $V(G_v)$ will not increase.

THEOREM 6 *The total distance of the d-dimensional k-ary hypercube tree $T_k(d)$ is minimum over all spanning trees of $H_k(d)$.*

Proof We use induction on d. If $d = 1$, then, clearly, $T_k(d) = K_{1,k-1}$, the star graph of order k, which is optimal. For the induction step, suppose that $d \geq 2$, and suppose that the k-ary hypercube tree $T_k(d-1)$ has minimum total distance over all spanning trees $G_k(d-1)$ for the $(d-1)$-dimensional k-ary hypercube $H_k(d-1)$. Let $G_k(d)$ be any spanning tree for the k-ary hypercube $H_k(d)$. By Lemma 6, we may assume that $G_k(d)$ is good, so it has k subtrees $G_k^i(d-1)$, $0 \leq i < k$. By repeated application of Lemma 7, we may also assume that $G_k(d)$ has no cross edges. Let

$$\sigma_1(G_k(d)) = \sum_{0 \leq i < j < k} \sum_{u \in G_k^i(d-1)} \sum_{v \in G_k^j(d-1)} d_{G_k(d)}(u, v)$$

and

$$\sigma_2(G_k(d)) = \sum_{i=0}^{k-1} \sigma(G_k^i(d-1)).$$

Define $\sigma_1(T_k(d))$ and $\sigma_2(T_k(d))$ similarly. So, $\sigma(G_k(d)) = \sigma_1(G_k(d)) + \sigma_2(G_k(d))$, and $\sigma(T_k(d)) = \sigma_1(T_k(d)) + \sigma_2(T_k(d))$. Since $G_k(d)$ has no cross edges, each subtree spans exactly one subcube, and hence the number of nodes in each subtree of $G_k(d)$ is exactly k^{d-1}. Thus, we have

$$\begin{aligned}
\sigma_1(G_k(d)) &= \sum_{0\le i<j<k} \sum_{u\in G_k^i(d-1)} \sum_{v\in G_k^j(d-1)} d_{G_k(d)}(u,v) \\
&= \sum_{0\le i<j<k} \sum_{u\in G_k^i(d-1)} \sum_{v\in G_k^j(d-1)} (d(u, i0^{d-1}) + d(j0^{d-1}, v)) + k^{2d-2} \sum_{0\le i<j<k} l_{i,j} \\
&= \sum_{0\le i<j<k} k^{d-1}(d_{G_k^i(d-1)}(i0^{d-1}) + d_{G_k^j(d-1)}(j0^{d-1})) + k^{2d-2} \sum_{0\le i<j<k} l_{i,j} \\
&\ge \sum_{0\le i<j<k} k^{d-1}(d_{iH_k(d)}(i0^{d-1}) + d_{jH_k(d)}(j0^{d-1})) + k^{2d-2} \sum_{0\le i<j<k} l_{i,j}, \qquad (20)
\end{aligned}$$

where inequality (20) follows from Equation (19), and $l_{i,j}$ is as defined in the proof of Lemma 4. The derivation here is similar to that in the proof of Lemma 4, so after simplification, Equation (20) reduces to

$$\sigma_1(G_k(d)) \ge d(k-1)^2 k^{2d-2} = \kappa\sigma(H_k(d)) = \sigma_1(T_k(d)). \qquad (21)$$

By induction, for $0 \le i < k$, $\sigma(G_k^i(d-1)) \ge \sigma(T_k(d-1))$. Hence,

$$\sigma_2(G_k(d)) = \sum_{i=0}^{k-1} \sigma(G_k^i(d-1)) \ge k\sigma(T_k(d-1)) = \sigma_2(T_k(d)). \qquad (22)$$

Combining inequalities (21) and (22) yields

$$\sigma(G_k(d)) = \sigma_1(G_k(d)) + \sigma_2(G_k(d)) \ge \sigma_1(T_k(d)) + \sigma_2(T_k(d)) = \sigma(T_k(d)).$$

It follows that $\sigma(T_k(d))$ is minimum over all spanning trees of $H_k(d)$. ■

7. Average distance of the *k*-ary hypercube tree

THEOREM 7 *The average distance of the d-dimensional k-ary hypercube tree, $d \ge 1$, is*

$$\mu(T_k(d)) = \frac{2d(k-1)k^{d-1}}{k^d - 1} - \frac{2}{k}.$$

Proof By Theorem 6,

$$\begin{aligned}
\mu(T_k(d)) &= \binom{k^d}{2}^{-1} \left(2\sigma(H_k(d)) - \frac{2}{k}\binom{k^d}{2}\right) \\
&= 2\mu(H_k(d)) - \frac{2}{k} \qquad (23) \\
&= \frac{2d(k-1)k^{d-1}}{k^d - 1} - \frac{2}{k}.
\end{aligned}$$

■

Given a graph G, let $r(G) = \min\{\sigma(T)/\sigma(G) \mid T$ is a spanning tree of $G\}$. Entringer *et al.* [8] have shown that for a connected graph G of order n, $r(G) \leq 2(1 - 1/n)$, and equality is achieved if and only if $G = K_n$ and $T = K_{1,n-1}$. Dobrynin *et al.* [5] stated that the dependence of r on the density of G is not clear, and conjectured that if T is of minimum total distance over all possible spanning trees of $H_2(d)$, then

$$r(H_2(d)) = 2\left(1 - \frac{1}{d}\right) + \frac{1}{d2^{d-1}} \sim 2. \tag{24}$$

In Theorem 6, we proved that $\sigma(T_k(d))$ is of minimum total distance among all spanning trees of $H_k(d)$. Consequently, by Equation (23),

$$\begin{aligned} r(H_k(d)) &= 2 - 2/(k\mu(H_k(d))) \\ &= 2 - \frac{2(k^d - 1)}{d(k-1)k^d} \\ &= 2 - \frac{2}{d(k-1)} + \frac{2}{d(k-1)k^d} \\ &= 2\left(1 - \frac{1}{d(k-1)}\right) + \frac{2}{d(k-1)k^d}. \end{aligned} \tag{25}$$

By Equation (25), the limiting value of $r(H_k(d))$ is 2. Clearly, Equation (25) simplifies to Equation (24) when $k = 2$.

8. Conclusions and open problems

In this paper, we presented some of the structural properties of the d-dimensional k-ary hypercube tree $T_k(d)$ and investigated in detail its total (and hence average) distance. Aside from its applications, the derived formula for its total distance can be used to evaluate or assess the performance of approximation or randomized algorithms that are designed for general graphs, especially regular graphs. It is worthwhile noting that performing BFS on the generalized hypercube may not result in an optimal tree as shown in Figure 7. In this figure, two BFS spanning trees are shown for the graph $K_2 \times K_3$ (Figure 7(a)). The tree in Part (b) is optimal, while that in Part (c) is not.

This is due to the nondeterministic nature of BFS. Since $K_2 \times K_3$ is vertex-transitive (but not edge-transitive), Figure 7 also shows that performing BFS on vertex-transitive graphs may not result in optimal spanning trees. However, applying BFS on the family of odd graphs, which is a generalization of the Petersen graph, *may* result in optimal spanning trees. This is (only) an observation of the results of conducting simple experimentation. If this turns out to be true, then, perhaps, it is due to the fact that they are distance-transitive.

The foregoing discussion raises the interesting question of characterizing those graphs on which applying BFS always gives optimal spanning trees.

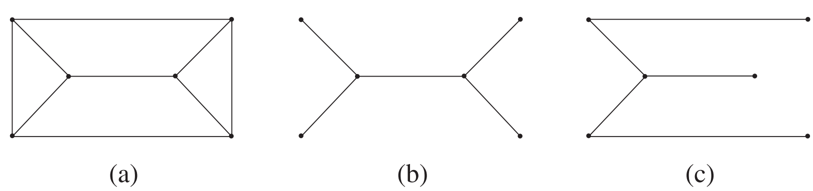

Figure 7. (a) A vertex-transitive graph. (b) Optimal BFS tree. (c) Nonoptimal BFS tree.

Acknowledgements

The author is grateful to KFUPM for their continual support. He would especially like to thank the anonymous reviewers for their many helpful comments, suggestions and corrections that significantly improved the presentation. Thanks to the editor for the post-comments.

References

[1] M. Aboelaze, *MLH: A hierarchical hypercube network*, Networks 28 (1996), pp. 157–165.
[2] M.H. Alsuwaiyel, *On the average distance of the hypercube tree*, Int. J. Comput. Math. 87 (2008), pp. 1208–1216.
[3] L.N. Bhuyan and D.P. Agrawal, *Generalized hypercube and hyperbus structures for a computer network*, IEEE Trans. Comput. 33 (1984), pp. 323–333.
[4] F. Buckley and F. Harary, *Distance in Graphs*, Addison-Wesley, Reading, MA, 1990.
[5] A.A. Dobrynin, R. Entringer, and I. Gutmann, *Wiener index of trees: Theory and applications*, Acta Appl. Math. 66 (2001), pp. 211–249.
[6] T. Dong, *A linear time pessimistic one-step diagnosis algorithm for hypercube multicomputer systems*, Parallel Comput. 31 (2005), pp. 933–947.
[7] J.K. Doyle and J.E. Graver, *Mean distance in a graph*, Discrete Math. 17 (1977), pp. 147–154.
[8] R.C. Entringer, D.J. Kleitman, and L.A. Székely, *A note on spanning trees with minimum average distance*, Bull. Inst. Combin. Appl. 17 (1996), pp. 71–78.
[9] I. Gutman and O.E. Polansky, *Mathematical Concepts in Organic Chemistry*, Springer Verlag, Berlin, 1984.
[10] C.-T. Ho and S. Johnson, *Distributed routing algorithm for broadcasting and personalized communication in hypercubes*, in *Proceedings of the IEEE 1986 International Conference on Parallel Processing, St. Charles, Ilinois, USA, August 19–22, 1986*, K. Hwang, S.M. Jacobs, and E.E. Swartzlander, eds., IEEE Computer Society Press, Washington, DC, 1986, pp. 640–648.
[11] S. Johnson and C.T. Ho, *Optimum broadcasting and personalized communication in hypercubes*, IEEE Trans. Comput. 38 (1989), pp. 1249–1268.
[12] C. Jordan, *Sur les assemblage des lignes*, J. Reine Angew. Math. 70 (1869), pp. 185–190.
[13] J.W. Moon, *On the total distance between nodes in trees*, Syst. Sci. Math. Sci. 9 (1996), pp. 93–96.
[14] M. Peercy and P. Banerjee, *Software schemes of reconfiguration and recovery in distributed memory multicomputers using the actor model*, in *FTCS '95 (FTCS-25) Proceedings of the Twenty-Fifth International Symposium on Fault-Tolerant Computing, Pasadena, California, USA, June 27–30, 1995*, IEEE Computer Society Press, Washington, DC, 1995, pp. 479–488. Available at http://www.informatik.uni-trier.de/∼ley/db/conf/ftcs/index.html.
[15] J. Plesnik, *On the sum of all distances in a graph or digraph*, J. Graph Theory 8 (1984), pp. 1–21.
[16] M. Tchuente, P. Melatagia Yonta, Y. Jean-Michel Nlong II, and Y. Denneulin, *On the minimum average distance spanning tree of the hypercube*, Acta Appl. Math. (2008), pp. 219–236.
[17] H. Wiener, *Structural determination of paraffin boiling points*, J. Am. Chem. Soc. 69 (1947), pp. 17–20.
[18] B.Y. Wu, K.M. Chao, and C.Y. Tang, *Light graphs with small routing cost*, Networks 39 (2002), pp. 130–138.
[19] B. Zelinka, *Medians and peripherians of trees*, Arch. Math. (Brno) 4 (1968), pp. 87–95.