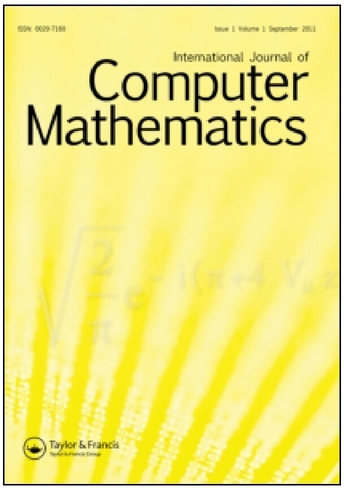

International Journal of Computer Mathematics
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713455451

On the average distance of the hypercube tree

M. H. Alsuwaiyel[a]
[a] Department of Information and Computer Science, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia

First published on: 14 October 2008

To cite this Article Alsuwaiyel, M. H.(2010) 'On the average distance of the hypercube tree', International Journal of Computer Mathematics, 87: 6, 1208 — 1216, First published on: 14 October 2008 (iFirst)
To link to this Article: DOI: 10.1080/00207160802195415
URL: http://dx.doi.org/10.1080/00207160802195415

International Journal of Computer Mathematics
Vol. 87, No. 6, May 2010, 1208–1216

Taylor & Francis
Taylor & Francis Group

On the average distance of the hypercube tree

M.H. Alsuwaiyel*

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

(*Received 17 November 2006; revised version received 29 February 2008; accepted 21 March 2008*)

Given a graph G on n vertices, the total distance of G is defined as $\sigma G = (1/2)\sum_{u,v\in V(G)} d(u,v)$, where $d(u,v)$ is the number of edges in a shortest path between u and v. We define the d-dimensional hypercube tree T_d and show that it has a minimum total distance $\sigma(T_d) = 2\sigma(H_d) - \binom{n}{2} = (dn^2/2) - \binom{n}{2}$ over all spanning trees of H_d, where H_d is the d-dimensional binary hypercube. It follows that the average distance of T_d is $\mu(T_d) = 2\mu(H_d) - 1 = d(1 + 1/(n-1)) - 1$.

Keywords: average distance; total distance; average delay; wiener index; hypercube tree

2000 AMS Subject Classification: 05C12; 05C05; 05C85; 05C90

1. Introduction

Let $G = (V, E)$ be a connected undirected graph with $|V(G)| = n$. The order of G is n. 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 $u, v \in V(G)$, the distance of v is defined as

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

The total distance of the graph G is defined as

$$\sigma(G) = \frac{1}{2}\sum_{v\in V(G)} d_G(v),$$

that is, the sum of distances between all unordered pairs. The average distance is defined as

$$\mu(G) = \frac{1}{\binom{n}{2}}\sigma(G).$$

*Email: suwaiyel@kfupm.edu.sa

ISSN 0020-7160 print/ISSN 1029-0265 online

DOI: 10.1080/00207160802195415
http://www.informaworld.com

The average distance, also known as transmission delay, is one of the most important measures of the efficiency of an interconnection network modelled by a graph. The diameter of a graph, which is the maximum node-to-node distance, is one of the factors taken into account when investigating a communication network. However, these pairs of nodes realizing the diameter may account for only a small fraction of the total number of pairs. Therefore, the average distance may be a more effective measure of the average performance of a network than its diameter, as it is an indicator for the expected travel time between two randomly chosen points of the network.

The average distance has been investigated by several authors and under different names, such as mean distance [6], total distance [14], transmission [15], total routing cost [20], and the Wiener index [3,19], with the latter being the oldest and most common. Given a network, which is modelled by a graph G, it may be possible to replace G by a subgraph H of G without significantly affecting the quality of communication. In this work, it is shown that in the case of the hypercube network, using the hypercube tree instead (which is defined in Section 3), it is possible to reduce the number of links by a factor of $\log n$ at the expense of increasing the average distance by a factor less than 2.

The algorithmic aspects of the average distance are investigated in [4,6]. In general, when the graph is weighted, finding a spanning tree with minimum average distance (or total distance), also called an MAD, is NP-hard [12]. Entringer *et al.* [7] showed that there is a spanning tree whose average distance is less than twice the average distance of the original, and that such a tree can be found in polynomial time. The Wiener index, defined as σG, was originally introduced by Harold Wiener [19] in 1947 and has numerous applications in physical chemistry [13]. It has been extensively studied (see [3] for an excellent survey and results).

The hypercube tree, which will be defined in Section 3, is known in the literature as the 'spanning binomial tree' (SBT) mostly in the context of communication and broadcasting in the hypercube [1,8,10]. The names 'completely unbalanced spanning tree' (CUST) [21] and 'hypercube tree' also appeared in the contexts of fault-tolerant computing and diagnosis of hypercube multicomputer systems to isolate faulty processors [5].

Broadcasting and personalized communication in a hypercube is done by constructing an SBT with a root at the source node and following the links of this tree to broadcast the message to all the nodes [8,10]. In [1], the same strategy (with some modification) was used for broadcasting in the multilevel hierarchical hypercube network MLH.

Distributed-memory hypercube computers are exposed to faults at the node and edge levels, which result in significant performance degradation. Expensive approaches were proposed to improve the fault tolerance of hypercube networks by using spares or by reconfiguration [17,18], such as the use of spare links and nodes [2], augmenting each node with one extra node [9], the use of multiple virtual nodes on each node for workload redistribution under faults [16], or reconfiguring the run-time system [17] in the case of faults. For a fixed number of nodes, the CUST used in [18] requires much less number of edges than a hypercube. When the number of faulty edges and their distribution still allow a tree to be formed in hypercube, reconfiguring the running application to a tree provides a continuation scheme in the presence of faults. In other words, a running hypercube application may switch to a tree-like reconfiguration in the presence of faulty edges. This leads to a smooth degradation in application throughput as the network performance is only twice that of the original hypercube. So, the tree presents a reconfiguration scheme for improving hypercube resilience to faulty edges.

2. Preliminaries

The eccentricity of a node v, denoted by $\mathrm{ecc}(v)$, in a connected graph G is the length of a longest of all shortest paths between v and every other node in G. The maximum eccentricity is the graph diameter. The minimum graph eccentricity is called the graph radius, denoted by $\rho(G)$. The centre C of a graph is the set of vertices of graph eccentricity equal to the graph radius (also called the set of central points). A branch B of a tree T at a vertex 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 vertex 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 vertices in a connected component of $T - v$. A centroid of a tree T is the set of vertices of T with minimum branch weight. The following theorem is due to Jordan [11].

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* $\mathrm{bw}(c_0) = \mathrm{bw}(c_1) = n/2$. *In both cases, if* $v \in V(T) - \mathcal{C}$, *then* $\mathrm{bw}(v) \geq n/2$.

Zelinka [21] *characterized the set of vertices with minimum distance in a tree.*

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

3. The hypercube tree

The (binary) d-dimensional hypercube H_d, $d \geq 0$, is defined as an undirected graph with $n = 2^d$ vertices and $dn/2 = d2^{d-1}$ edges. The vertices are labelled with all elements in $\{b_1 b_2 \cdots b_d | b_i \in \{0, 1\}\}$, and there is an edge between two vertices u and v if and only if u and v differ in exactly one bit. The left subcube or 0-cube $0H_d$ is the induced subgraph of H_d on $\{0b_1 b_2 \cdots b_{d-1} | b_i \in \{0, 1\}\}$. Similarly, the right subcube or 1-cube $1H_d$ is the induced subgraph of H_d on $\{1b_1 b_2 \cdots b_{d-1} | b_i \in \{0, 1\}\}$.

For $d = 1, 2, \ldots$, we define the d-dimensional hypercube tree rooted at vertex $00 \cdots 0$ (d zeros), which we denote by T_d, as a rooted tree whose set of vertices is $V(H_d)$, and whose set of edges $E(H_d)$ is constructed using one of the following two construction methods (see Figure 1 for an example).

(1) Recursive: If $d = 1$, then $E(T_1) = \{(0, 1)\}$. Otherwise, T_d is constructed recursively by linking the roots of two copies of T_{d-1} by an edge and designating one of its two ends as the root. That is,

$$E(T_d) = \{(0u, 0v) | (u, v) \in E(T_{d-1})\} \cup \{(1u, 1v) | (u, v) \in E(T_{d-1})\} \cup \{(0^d, 10^{d-1})\}.$$

(2) Iterative: If $d = 1$, then $E(T_1) = \{(0, 1)\}$. Otherwise, T_d is constructed from T_{d-1} by attaching a leaf node to each vertex in T_{d-1}. That is,

$$E(T_d) = \{(u0, v0) | (u, v) \in E(T_{d-1})\} \cup \{(v0, v1) | v \in V(T_{d-1})\}.$$

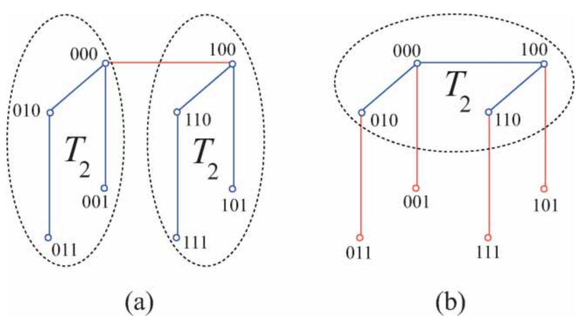

Figure 1. Construction of T_3 from T_2. (a) recursive. (b) iterative.

It should be noted that T_d can be constructed by applying ordinary breadth-first traversal on H_d starting from vertex 0^d. However, using BFS costs $\Theta(dn) = \Theta(n \log n)$, while direct construction costs only $\Theta(n)$.

Similar to the hypercube, we define the left subtree or 0-tree $0T_d$ as the induced subtree of T_d on $V(0H_d)$ and the right subtree or 1-tree $1T_d$ as the induced subtree of T_d on $V(1H_d)$. In other words, $0T_d$ is T_{d-1} with every label prefixed with 0 and $0T_d$ is T_{d-1} with every label prefixed with 1. For brevity, we will call a vertex even (odd) if its label is the binary representation of an even (odd) integer.

THEOREM 3 *For $d = 1, 2, \ldots$, let T_d and T'_d be two trees obtained using the iterative and recursive construction methods, respectively. Then, $T_d = T'_d$.*

Proof Fix a vertex $v = x_1x_2 \cdots x_{j-1}100 \cdots 0$ (different from the root $00 \cdots 0$), where $|v| = d$, and $x_i \in \{0, 1\}$, $1 \le i < j \le d$. We show in both constructions that

$$p(v) = x_1x_2 \cdots x_{j-1}000 \cdots 0$$

is the parent of v. By construction of T_d, when v was first created at possibly some earlier stage of the construction, its label was of the form $x_1x_2 \cdots x_{j-1}1$ and the label of its parent was of the form $x_1x_2 \cdots x_{j-1}0$. From that point on, a zero would be appended to the labels of both v and $p(v)$. This proves the parental relationship assertion for T_d.

We now prove the assertion for T'_d. To this end, rewrite v as

$$v = y_{j-1}y_{j-2} \cdots y_1100 \cdots 0.$$

We show, by induction on d, that the parent of v in T'_d is

$$p(v) = y_{j-1}y_{j-2} \cdots y_1000 \cdots 0.$$

Let $y_0 = 1$ in the definition of v and $y_0 = 0$ in the definition of $p(v)$. When $d = 1$, $v = 1$ and the parent of vertex 1 is 0. So assume that $d \ge 2$. Let u' and v' be two vertices in T'_{d-1}, where

$$u' = y_{j-2}y_{j-3} \cdots y_1000 \cdots 0 \quad \text{and} \quad v' = y_{j-2}y_{j-3} \cdots y_1100 \cdots 0;$$

both of length $d - 1$. By induction, $u' = p(v')$. To construct T'_d from T'_{d-1}, the edge (u', v') will be doubled: one copy will belong to the 0-tree of T'_d, in which case the labels of both u' and v' are prefixed with 0, and the other copy will belong to the 1-tree of T'_d, in which case the labels of both u' and v' are prefixed with 1. Hence, in T'_d, if $y_{j-1} = 0$, then

$$p(0y_{j-2}y_{j-3} \cdots y_1100 \cdots 0) = (0y_{j-2}y_{j-3} \cdots y_1000 \cdots 0),$$

and, if $y_{j-1} = 1$, then

$$p(1y_{j-2}y_{j-3} \cdots y_1100 \cdots 0) = (1y_{j-2}y_{j-3} \cdots y_1000 \cdots 0).$$

We conclude that the parental relationship assertion is true for T'_d too.

Now, since v is arbitrary, it follows that the parent of any vertex other than the root is the same in both trees T_d and T'_d. This, in turn, implies the natural isomorphism $\phi : V(T) \to V(T')$ defined by $\phi(v) = v$ for all v in $\{0, 1\}^d$, from which we conclude that $T_d = T'_d$. ■

4. Computing the total distance

First, we compute the distance of the root of $T_d d_{T_d}(0^d) = \sum_{w \in V(T_d)} d_{T_d}(0^d, w)$, and establish some relationship between distances in the hypercube tree T_d and its corresponding hypercube graph H_d.

LEMMA 1 *Let T_d be a d-dimensional hypercube tree. Then,*

(i) $\forall v \in V(T_d)\ d_{T_d}(0^d, v) = d_{H_d}(0^d, v)$;
(ii) $d_{T_d}(0^d) = d_{H_d}(0^d) = dn/2$.

Proof (i) The proof is by induction on $d \geq 1$. For $d = 1$, it is true, so suppose that $d \geq 2$. Observe that, by construction, for any vertex $0u$ in the 0-tree $0T_d$,

$$\begin{aligned} d_{T_d}(0^d, 0u) &= d_{0T_d}(0^d, 0u) \\ &= d_{T_{d-1}}(0^{d-1}, u) && (1) \\ &= d_{H_{d-1}}(0^{d-1}, u) \text{ (by induction)} \\ &= d_{H_d}(0^d, u), && (2) \end{aligned}$$

and for any vertex $1v$ in the 1-tree $1T_d$,

$$\begin{aligned} d_{T_d}(0^d, 1v) &= 1 + d_{1T_d}(10^{d-1}, 1v) \\ &= 1 + d_{T_{d-1}}(0^{d-1}, v) && (3) \\ &= 1 + d_{H_{d-1}}(0^{d-1}, v) \text{ (by induction)} \\ &= d_{H_d}(0^d, 1v). && (4) \end{aligned}$$

Hence, we conclude that

$$\forall v \in V(T_d)\ d_{T_d}(0^d, v) = d_{H_d}(0^d, v).$$

(ii) Since there are $n/2$ distances from 0^d to vertices in the 0-tree and $n/2$ distances from 0^d to vertices in the 1-tree, which additionally contribute $n/2$ to the total distance, and by Equations (1) and (3), $d_{T_d}(0^d)$ can be expressed by the recurrence

$$d_{T_d}(0^d) = \begin{cases} 1 & \text{if } d = 1, \\ 2d_{T_{d-1}}(0^{d-1}) + (n/2) & \text{if } d > 1, \end{cases}$$

whose solution is $d_{T_d}(0^d) = dn/2$. By part (i), $d_{T_d}(0^d) = d_{H_d}(0^d) = dn/2$. ■

THEOREM 4 *The total distance of the hypercube tree T_d is*

$$\sigma(T_d) = 2\sigma(H_d) - \binom{n}{2} = \frac{dn^2}{2} - \binom{n}{2},$$

which is minimal over all spanning trees of H_d.

Proof First, note that by Lemma 1 and the symmetry of the hypercube graph,

$$\sigma(H_d) = \frac{n}{2}\, d_{H_d}(0^d) = \frac{dn^2}{4}. \tag{5}$$

Next, we find the total distance between all vertices in the 0-tree and all vertices in the 1-tree. Let $u \in 0T_d$ and $v \in 1T_d$. Then, we have (Figure 2)

$$d_{T_d}(u, v) = d_{0T_d}(u, 0^d) + (1 + d_{1T_d}(10^{d-1}, v)).$$

Using Equations (2) and (4), we obtain,

$$d_{T_d}(u, v) = d_{H_d}(0^d, u) + d_{H_d}(0^d, v). \tag{6}$$

Summing over all vertices $u \in 0T_d$ and $v \in 1T_d$ yields

$$\begin{aligned}\sum_{u\in 0T_d}\sum_{v\in 1T_d} d_{T_d}(u, v) &= \sum_{u\in 0H_d}\sum_{v\in 1H_d}\left(d_{H_d}(0^d, u) + d_{H_d}(0^d, v)\right) \\ &= \frac{n}{2}\sum_{u\in 0H_d} d_{H_d}(0^d, u) + \frac{n}{2}\sum_{v\in 1H_d} d_{H_d}(0^d, v) \\ &= \frac{n}{2}\sum_{w\in H_d} d_{H_d}(0^d, w) \\ &= \frac{n}{2}\, d_{H_d}(0^d) \\ &= \sigma(H_d),\end{aligned} \tag{7}$$

where the last equality follows from Equation (5). Since $\sigma(T_d)$ is the sum of total distances in the 0-tree, 1-tree, and the total distance between all vertices in the 0-tree and all vertices in the 1-tree, $\sigma(T_d)$ can be expressed by the recurrence

$$\sigma(T_d) = \begin{cases} 1 & \text{if } d = 1, \\ 2\sigma(T_{d-1}) + \sigma(H_d) & \text{if } d > 1. \end{cases}$$

whose solution is

$$\sigma(T_d) = 2\sigma(H_d) - \binom{2^d}{2},$$

and, by Equation (5),

$$\sigma(T_d) - \frac{d2^{2d}}{2} - \binom{2^d}{2}.$$

Finally, note that, by Equation (7), the total distance between vertices in $0T_d$ and $1T_d$, whose paths must pass through the centroid, is minimum. Hence, if we assume that $\sigma(T_{d-1})$ is minimum, then it follows by induction that $\sigma(T_d)$ is also of minimum value. ∎

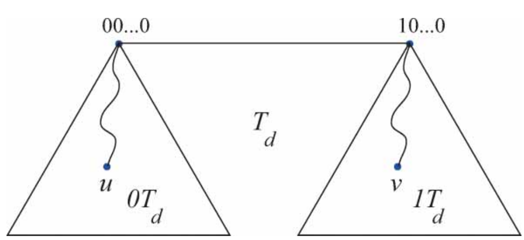

Figure 2. Proof of Theorem 4.

Theorem 4 gives rise to the following sequence for $\sigma(T_d)$, $d = 1, 2, \ldots$

$$1, 10, 68, 392, 2064, 10272, \ldots.$$

Let $T = (V, E)$ be a tree and $e = (u, v)$ be an edge of T. Let $n_u(e)$ denote the number of vertices of T lying closer to u than to v, and let $n_v(e)$ denote the number of vertices of T lying closer to v than to u. The following theorem was discovered by Wiener in 1947 [3,19].

THEOREM 5 *Let $T = (V, E)$ be a tree. Then, $\sigma(T) = \sum_{e\in E(T)} n_u(e)n_v(e)$.*

Define the weight of an edge e as $w(e) = n_u(e)n_v(e)$. For $1 \le j \le d$, let $E_j(T_d)$ denote the set of edges in $E(T_d)$ with weight $n/2^j(n - (n/2^j))$.

PROPOSITION 1 *Let T_d be a d-dimensional hypercube tree. Then,*

(i) $E(T_d) = E_1(T_d) \cup E_2(T_d) \cup \cdots \cup E_d(T_d)$, *and* $|E_j(T_d)| = 2^{j-1}$, $1 \le j \le d$;
(ii) $\sigma(T_d) = \sum_{j=1}^{d} 2^{j-1} n/2^j(n - n/2^j)$.

Proof (i) If $d = 1$, then there is exactly one edge with weight 1, so suppose $d \ge 2$. Assume inductively that

$$E(T_{d-1}) = \bigcup_{j=1}^{d-1} E_j(T_{d-1}), \quad \text{and} \quad |E_j(T_{d-1})| = 2^{j-1}, \ \ 1 \le j \le d-1.$$

Let $e = (u, v)$ be an edge in $E_j(T_{d-1})$ for some j, $1 \le j \le d-1$. By construction, T_d is obtained from T_{d-1} by attaching a leaf node to each vertex of T_{d-1}. Hence, both $n_u(e)$ and $n_v(e)$ are doubled, which means that $E(T_d)$ contains exactly 2^{j-1} edges with weight

$$2 \times 2 \times \frac{n/2}{2^j}\left(\frac{n}{2} - \frac{n/2}{2^j}\right) = \frac{n}{2^j}\left(n - \frac{n}{2^j}\right).$$

Since $e = (u, v)$ is arbitrary, we conclude that $|E_j(T_d)| = 2^{j-1}$ for $1 \le j \le d-1$. Moreover, there will be $n/2$ additional edges in T_d with weight $n - 1$, that is, $|E_d(T_d)| = 2^{d-1}$. It follows that $E(T_d) = \bigcup_{j=1}^{d} E_j(T_d)$, and for $1 \le j \le d$, $|E_j(T_d)| = 2^{j-1}$.

(ii) Follows from (1) and Theorem 5. ■

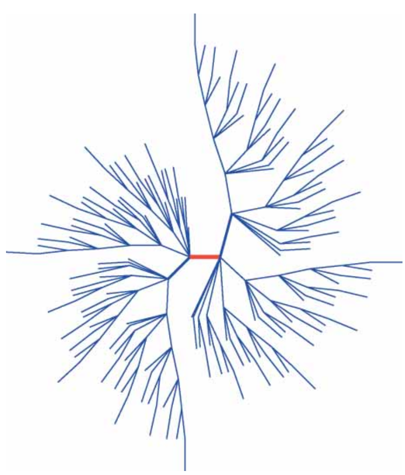

Figure 3. The horizontal edge in the middle is the central edge of I_8.

As illustrated in Figure 3, there is an edge (the central edge) with weight $(n/2)^2$, two edges with weight $(n/4)(3n/4)$, and in general 2^{j-1} edges with weight $n/2^j(n-(n/2^j))$. In this figure, the horizontal edge in the middle is the central edge of T_8.

5. Mean distance

The average distance of the hypercube of dimension $d \geq 1$ is computed as

$$\begin{aligned}\mu(H_d) &= \frac{1}{\binom{n}{2}}\sigma(H_d)\\ &= \frac{dn^2}{4\binom{n}{2}}\\ &= \frac{dn}{2(n-1)}\\ &= \frac{d}{2}\left(1+\frac{1}{n-1}\right).\end{aligned}$$

Similarly, the average distance of the hypercube tree of dimension $d \geq 1$ is computed as

$$\begin{aligned}\mu(T_d) &= \frac{1}{\binom{n}{2}}\sigma(T_d)\\ &= \frac{1}{\binom{n}{2}}\left(2\sigma(H_d)-\binom{n}{2}\right)\\ &= 2\mu(H_d)-1\\ &= d\left(1+\frac{1}{n-1}\right)-1.\end{aligned} \tag{8}$$

Hence, we have the following.

THEOREM 6 *The average distance of the hypercube tree is* $\mu(T_d) = d(1+1/(n-1))-1$.

6. Conclusion

Given a graph G, let $s(G) = \min\{\sigma(T)/\sigma(G) | T$ is a spanning tree of $G\}$. In [7], Entringer *et al.* have shown that for a connected graph G of order n, $s(G) \leq 2(1-1/n)$, and equality is achieved if and only if $G = K_n$ and $T = K_{1,n-1}$. In [3], Dobrynin *et al.* stated that the dependence of s 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_d, then

$$s(H_d) = 2\left(1-\frac{1}{d}\right)+\frac{1}{d2^{d-1}} \sim 2. \tag{9}$$

In Theorem 4, we proved that $\sigma(T_d)$ is of minimum total distance among all spanning trees of H_d. Consequently, by Equation (8),

$$s(H_d) = 2 - \frac{1}{\mu(H_d)} = 2 - \frac{2(n-1)}{dn} = 2\left(1 - \frac{1}{d}\right) + \frac{1}{d2^{d-1}},$$

and $s(H_d)$ has 2 as its limiting value.

References

[1] M. Aboelaze, *MLH: A hierarchical hypercube network*, Networks 28 (1996), pp. 157–165.
[2] P. Banerjee, J.T. Rahmeh, C.B. Stunkel, V.S.S. Nair, K. Roy, and J.A. Abraham, *An evaluation of system-level fault tolerance on the Intel hypercube multiprocessor*, Proc. 18th Int. Symp. Fault-Tolerant Comput. 1988, pp. 362–367.
[3] A.A. Dobrynin, R. Entringer, and I. Gutmann, *Wiener index of trees: theory and applications*, Acta Appl. Math. 66 (2001), pp. 211–249.
[4] P. Dankelmann, *Computing the average distance of an interval graph*, Inf. Process Lett. 43 (1993), pp. 311–314.
[5] T. Dong, *A linear time pessimistic one-step diagnosis algorithm for hypercube multicomputer systems*, Parallel Comput. 31 (2005), pp. 933–947.
[6] J.K. Doyle and J.E. Graver, *Mean distance in a graph*, Discrete Math. 17 (1977), pp. 147–154.
[7] R.C. Entringer, D.J. Kleitman, and Székely, *A note on spanning tees with minimum average distance*, Bull. Inst. Combin. Appl. 17 (1996), pp. 71–78.
[8] C.T. Ho and S. Johnson, *Distributed routing algorithm for broadcasting and personalized communication in hypercubes*, Proc. 1986 Int. Conf. Par. Proc. pp. 640–648.
[9] B.A. Izadi, F. Ozguner, and A. Acan, *Highly fault-tolerant hypercube multicomputer*, Proc. IEE Comput. Digit. Tech. 146 (1999), pp. 77–82.
[10] S. Johnson and C.T. Ho, *Optimum broadcasting and personalized communication in hypercubes*, IEEE Trans. Comput. 38 (1989), pp. 1249–1268.
[11] C. Jordan, *Sur les assemblage des lignes*, J. Reine Angew. Math. 70 (1869), pp. 185–190.
[12] D.S. Johnson, J.K. Lenstra, and A.H.G.R. Kan, *The complexity of the network design problem*, Networks 8 (1978), pp. 279–285.
[13] I. Gutman and O.E. Polansky, *Mathematical Concepts in Organic Chemistry*, Springer Verlag, Berlin, 1984.
[14] J.W. Moon, *On the total distance between nodes in trees*, Systems Sci. Math. Sci. 9 (1996), pp. 93–96.
[15] J. Plesnik, *On the sum of all distances in a graph or digraph*, J. Graph Theory 8 (1984), pp. 1–21.
[16] M. Peercy and P. Banerjee, *Design and analysis of software reconfiguration strategies for hypercube multicomputers under multiple faults*, Proc. 22nd Int. Symp. Fault-Tolerant Comput. 1992, pp. 448–455.
[17] ———, *Software schemes of reconfiguration and recovery in distributed memory multicomputers using the actor model*, Proc. 25th Int. Symp. Fault-Tolerant Comput. 1995, pp. 479–488.
[18] K.M. Al-Tawil and D.R. Avresky, *An effective approach for achieving fault tolerance in hypercubes*, Fault-Tolerant Parallel Distribut. Syst. 12–14, 1994, pp. 113–120.
[19] H. Wiener, *Structural determination of paraffin boiling points*, J. Am. Chem. Soc. 69 (1947), pp. 1–24.
[20] B.Y. Wu, K.M. Chao, and C.Y. Tang, *Light graphs with small routing cost*, Networks 39 (2002), pp. 130–138.
[21] B. Zelinka, *Medians and peripherians of trees*, Arch. Math. (Brno) 4 (1968), pp. 87–95.