Constructing Two Completely Independent Spanning Trees in Balanced Hypercubes

Yi-Xian YANG†(a), Kung-Jui PAI†(b), Ruay-Shiung CHANG†(c), Nonmembers, and Jou-Ming CHANG†(d), Member

SUMMARY A set of spanning trees of a graphs G are called completely independent spanning trees (CISTs for short) if for every pair of vertices x, y ∈ V(G), the paths joining x and y in any two trees have neither vertex nor edge in common, except x and y. Constructing CISTs has applications on interconnection networks such as fault-tolerant routing and secure message transmission. In this paper, we investigate the problem of constructing two CISTs in the balanced hypercube BH_n, which is a hypercube-variant network and is superior to hypercube due to having a smaller diameter. As a result, the diameter of CISTs we constructed equals to 9 for BH_2 and 6n – 2 for BH_n when n ≥ 3.

key words: interconnection networks, completely independent spanning trees, balanced hypercubes, diameter

1. Introduction

Interconnection networks are usually modeled as undirected simple graphs G = (V,E), where the vertex set V (= V(G)) and the edge set E (= E(G)) represent the set of processors and the set of communication channels between processors, respectively. A spanning tree T of G is an acyclic connected subgraph of G such that V(T) = V(G). A vertex in a tree is called a leaf if it has degree 1, and an inner-vertex otherwise. Two spanning trees T_1 and T_2 are edge-disjoint if E(T_1) ∩ E(T_2) = ∅, and are inner-vertex-disjoint if the paths joining any two vertices x and y in both trees have no vertex in common except for x and y. A set of k (≥ 2) spanning trees of G are called completely independent spanning trees (CISTs for short) if they are inner-vertex-disjoint (and are indeed edge-disjoint, see Theorem 1 in Sect. 2).

For interconnection networks, constructing multiple CISTs has diverse applications such as fault-tolerant routing and secure message transmission. Hasunuma [12] showed that determining whether a graph G admits CISTs is NP-complete, even for k = 2. P´eterfalvi [24] showed that, for any k ≥ 2, there exists k-connected graph which does not possess two CISTs. Accordingly, researches investigating sufficient conditions for graphs that admit multiple CISTs, such as degree-based conditions, can be found in [1], [3], [7], [15], [17]. Also, with the help of constructions, it has been confirmed that certain classes of graphs possess two CISTs, e.g., 4-connected maximal planar graphs [13], Cartesian product of any 2-connected graphs [14], 4-regular chordal rings [2], [23], crossed cubes [5], and several hypercube-variant networks [21]. In addition, more graphs possessing multiple CISTs can be found in [6], [12], [16], [19], [20], [22].

In this paper, we investigate the problem of constructing two CISTs in balanced hypercubes (defined later in Sect. 2), which were first proposed by Huang and Wu [18]. The balanced hypercube is also a hypercube-variant network and is superior to hypercube due to having a smaller diameter. Basic properties of balanced hypercubes have been acquired, such as bipartite graphs [18], vertex-transitivity [25], and edge-transitivity [26]. A particular property of the balanced hypercube is that each processor has a backup processor that shares the same neighborhood. For more results about the balanced hypercubes, please refer to [4], [8]–[11].

2. Preliminaries

The following two characterizations are important for studying CISTs.

**Theorem 1:** (Hasunuma [12]) A set of spanning trees T_1, T_2, ..., T_k are CISTS in a graph G = (V,E) iff they are edge-disjoint and for any v ∈ V, there is at most one spanning tree T_i such that v is an inner-vertex.

**Theorem 2:** (Arak [1]) A graph G = (V,E) admits k CISTs iff there is a partition of V into V_1, V_2, ..., V_k, which is called a k-CIST-partition, such that the following hold:

(i) for i ∈ {1, 2, ..., k}, the subgraph of G induced by V_i, denoted by G[V_i], is connected;

(ii) for distinct i, j ∈ {1, 2, ..., k}, the bipartite graph with bipartition V_i ∪ V_j and edge set {(x,y) ∈ E(G): x ∈ V_i, y ∈ V_j}, denoted by B(V_i, V_j, G), has no tree component.

For constructing two CISTs, the results in [5], [13], [14], [23] are based on Theorem 1, and the results in [2], [21] are based on Theorem 2. In this paper, we apply Theorem 1 to construct two CISTS of balanced hypercubes.

**Definition 1:** (Huang and Wu [18]) An n-dimensional balanced hypercube BH_n with n ≥ 1 consists of 2^{2n} vertices such that each vertex is labeled by an n-tuple...
(a_0, a_1, \ldots, a_{n-1}), where a_i \in \{0, 1, 2, 3\} for 0 \leq i \leq n - 1, and is adjacent to the following 2n vertices:

(i) (a_0 \pm 1, a_1, \ldots, a_{n-1});
(ii) (a_0 \pm 1, a_1, \ldots, a_{j-1}, a_j + (-1)^{a_0}, a_{j+1}, \ldots, a_{n-1})

for 1 \leq j \leq n - 1,

where arithmetics in the above labels are taken modulo 4.

From the above, we can see that the scalability of a balanced hypercube BH_n is growing at twice the traditional hypercube Q_n. For a labeled graph G, we denoted by G^i the graph obtained from G by appending the symbol i as the rightmost element in the n-tuple of each vertex. Wu and Huang[25] also showed that BH_n can be equivalently defined by the following recursive fashion.

Definition 2: BH_n can be recursively constructed by the following rules:

1. BH_1 is a 4-cycle (i.e., a cycle of length 4) with vertices labeled by (0), (1), and (2), respectively;
2. For n \geq 2, BH_n can be decomposed into four copies of BH_{n-1}, i.e., BH_{n-1}'_i for i \in \{0, 1, 2, 3\}, such that each vertex (a_0, a_1, \ldots, a_{n-2}, i) \in V(BH_{n-1}') is connected with two vertices:
   - (a_0 \pm 1, a_1, \ldots, a_{n-2}, i+1) \in V(BH_{n-1}') if a_0 is even.
   - (a_0 \pm 1, a_1, \ldots, a_{n-2}, i-1) \in V(BH_{n-1}') if a_0 is odd.

If u = (a_0, a_1, \ldots, a_{n-1}), the two adjacent vertices defined as above are called the out-neighbors of u. We denote by N^+(u) (resp. N^-(u)) the out-neighbor taken a_0 + 1 (resp. a_0 - 1) as its first element. Hereafter, for notational convenience, we omit to state "(mod 4)" in each element of an n-tuple and the superscript of a subgraph of the balanced hypercube. In fact, the decomposition of BH_n is more flexible. For vertices with label (a_0, a_1, \ldots, a_{n-1}) in BH_n, a subgraph decomposition can be made by fixing the same symbol at a particular element a_i (i \neq 0) of n-tuples. Figure 1 depicts BH_1, BH_2, and BH_3, where a set of edges drawn in thick lines in BH_3 indicates a subgraph isomorphic to BH_2 and vertices in the subgraph are with a_1 = 0.

Similarly, every hypercube-variant network possesses a recursive structure. For instance, denote by G_n and G_n^j for i \in \{0, 1\} the n-dimensional cube and the (n - 1)-dimensional cube with adding a symbol i at the highest position of the label for each vertex, respectively. Then G_n can be composed from G_{n-1}^0 and G_{n-1}^1 by adding appropriate edges in 2^{n-1} disjoint pairs of vertices between G_{n-1}^0 and G_{n-1}^1 (i.e., a perfect matching). Hence, Pai and Chang[21] proposed a unified approach to recursively construct two CISTs in several hypercube-variant networks, including hypercubes, locally twisted cubes, crossed cubes, parity cubes, and Möbius cubes. This unified approach is based on the following theorem.

Theorem 3: [21] For n \geq 5 and i \in \{0, 1\}, let T_j^0 and T_j^1 be two CISTs of G_{n-1}^i. Then, two CISTs of G_n, say 1 for j \in \{1, 2\}, can be constructed from T_j^0 and T_j^1 by adding an edge (u_j, v_j) \in E(G_n) to connect two inner-vertices u_j \in V(T_j^0) and v_j \in V(T_j^1).

In the above theorem, the two vertices u_j and v_j are called the port vertices of T_j^0 and T_j^1, respectively. The edge (u_j, v_j) is called the bridge of the construction.

3. Main Results

In this section, we present our constructing scheme of two CISTs in BH_n for n \geq 2. For a graph G, the diameter of G, denoted by diam(G), is the greatest distance between any
pair of vertices in \( G \). The center of \( G \) is defined to be the set of vertices which minimize the maximal distance from other vertices in \( G \). A well-known result is that the center of a tree \( T \) consists of either a singleton if \( \text{diam}(T) \) is even or two adjacent vertices otherwise.

Firstly, the two desired spanning trees of \( BH_2 \) are shown in Fig. 2. It is easy to check that \( T_1 \) and \( T_2 \) are edge-disjoint, and every vertex of \( BH_2 \) is a leaf either in \( T_1 \) or \( T_2 \). Thus, by Theorem 1, the two spanning trees are indeed CISTs of \( BH_2 \). Also, it is clear that \( \text{diam}(T_1) = \text{diam}(T_2) = 9 \), and \( \{(1,2),(2,1)\} \) and \( \{(0,3),(3,0)\} \) are the centers of \( T_1 \) and \( T_2 \), respectively. Therefore, we have the following lemma.

**Lemma 4:** \( BH_2 \) admits two CISTs with diameter 9.

In what follows, we consider the construction of two CISTs in the high-dimensional \( BH_n \) for \( n \geq 3 \) by recursion. Before this, we state a property similar to Theorem 3 and it is useful for constructing CISTs in \( BH_n \). This property can be proved by using the same proof technique of Theorem 3.

**Corollary 5:** For \( n \geq 3 \) and \( i \in \{0,1,2,3\} \), let \( T_i^1 \) and \( T_i^2 \) be two CISTs of \( BH_n \). Then, two CISTs of \( BH_n \), say \( T_j^i \) for \( j \in \{1,2\} \), can be constructed from \( T_i^0 \) and \( T_i^1 \) by adding three edges fulfilled one of the following conditions:

(i) For \( i \in \{0,1,2\} \), add \( (u,v) \in E(BH_n) \) to connect two inner-vertices \( u \in V(T_i^0) \) and \( v \in V(T_i^1) \). (ii) For \( i \in \{0,3,2\} \), add \( (u,v) \in E(BH_n) \) to connect two inner-vertices \( u \in V(T_i^1) \) and \( v \in V(T_i^0) \).

For instances, we consider \( BH_3 \) and \( BH_4 \) as follows. Note that, in \( BH_2 \), a particular vertex \( (1,1) \) is adjacent to the center vertex \( (2,1) \) in \( T_1 \), and a particular vertex \( (2,0) \) is adjacent to the center vertex \( (3,0) \) in \( T_2 \) (see Fig. 2 for vertices surrounded by dashed circles). Then, \( N''(2,0) = (0,1) \) and \( N''((3,0)) = (2,0, i - 1) \) in \( BH_3 \) for \( i \in \{0,1,2,3\} \). So, if we choose the pairs \( \{(2,0),(0,1)\} \) as port vertices to connect two trees \( T_1^0 \) and \( T_2^0 \) for \( i = 0,1,2 \), we obtain a spanning tree \( \hat{T}_1 \) in \( BH_3 \). Similarly, if we choose the pairs \( \{(3,0),(2,0, i - 1)\} \) as port vertices to connect two trees \( T_1^1 \) and \( T_2^1 \) for \( i = 0,1,2 \), then we obtain a spanning tree \( \hat{T}_2 \) in \( BH_3 \). See Fig. 3 for a schematic illustration, where each thick line indicates a bridge for connecting two port vertices. Also, a line segment indicates a path in the tree and a number attached to a line represents the length of the path. By Corollary 5, \( \hat{T}_1 \) and \( \hat{T}_2 \) are two CISTs of \( BH_3 \) and \( \text{diam}(\hat{T}_1) = \text{diam}(\hat{T}_2) = 16 \). Moreover, we observe that \( (1,1,2) \) is the unique center vertex of \( \hat{T}_1 \) and it is adjacent to a particular vertex \( (2,1,2) \). Also, \( (2,0,2) \) is the unique center vertex of \( \hat{T}_2 \) and it is adjacent to a particular vertex \( (3,2,0) \) (see Fig. 3 for vertices surrounded by dashed circles).

Then, using a similar approach, we can construct two CISTs, say \( \hat{T}_1 \) and \( \hat{T}_2 \), of \( BH_4 \) if we choose the pairs \( \{(1,1,2, i),(2,1,2, i - 1)\} \) as port vertices to connect two trees \( \hat{T}_1^0 \) and \( \hat{T}_2^0 \) for \( i = 0,3,2 \) and the pairs \( \{(2,0,2, i),(3,0,2, i + 1)\} \) as port vertices to connect two trees \( \hat{T}_1^1 \) and \( \hat{T}_2^1 \) for \( i = 0,1,2 \). See Fig. 4 for a schematic illustration.

In general, suppose that \( T_1 \) and \( T_2 \) are two constructed CISTs of \( BH_{n-1} \). For \( i \in \{0,1,2,3\} \), let \( T_i^1 \) and \( T_i^2 \) be two corresponding CISTs of \( BH_{n-1} \). If \( n \geq 3 \) is odd, we choose three pairs \( \{u,N(u)\} \) as port vertices to connect \( T_i^0 \) and \( T_i^{i+1} \) for \( i = 0,1,2 \), where \( u \in \{2,1,2,2,\ldots,2,i\} \) and \( N(u) = (1,1,2,2,\ldots,2,i+1) \). Then, we obtain a spanning tree \( \hat{T}_1 \) with the unique center vertex \( (1,1,2,2,\ldots,2,i) \).

Also, we choose pairs \( \{u,N(u)\} \) as port vertices to connect...
$T_i^2$ and $T_i^{2-1}$ for $i = 0, 3, 2$, where $u = (3, 0, 2, 2, \ldots, 2, i)$ and $N^-(u) = (2, 0, 2, 2, \ldots, i, i - 1)$. Then, we obtain a spanning tree $\hat{T}_2$ with the unique center vertex $(2, 0, 2, 2, \ldots, 2, 2)$.

On the other hand, if $n \geq 4$ is even, we choose three pairs $(u, N^+(u))$ as port vertices to connect $T_i^2$ and $T_i^{2+1}$ for $i = 0, 3, 2$, where $u = (1, 1, 2, 2, \ldots, 2, i)$ and $N^+(u) = (2, 1, 2, 2, \ldots, 2, i - 1)$. Then, we obtain a spanning tree $\hat{T}_1$ with the unique center vertex $(2, 1, 2, 2, \ldots, 2, 2)$. Also, we choose pairs $(u, N^+(u))$ as port vertices to connect $T_i^2$ and $T_i^{2+1}$ for $i = 0, 1, 2$, where $u = (2, 0, 2, 2, \ldots, 2, i)$ and $N^+(u) = (3, 0, 2, 2, \ldots, 2, i + 1)$. Then, we obtain a spanning tree $\hat{T}_2$ with the unique center vertex $(3, 0, 2, 2, \ldots, 2, 2)$.

By Corollary 5, the two spanning trees $\hat{T}_1$ and $\hat{T}_2$ are CISTS of $BH_n$. According to the recursive construction, it follows that
\[
\text{diam}(\hat{T}_i) = 2 \left( \left\lceil \frac{\text{diam}(T_i)}{2} \right\rceil + 3 \right) \text{ for } i \in \{1, 2\}.
\]

Then, solving the equation under the base case $\text{diam}(T_i) = 9$ when $n = 2$ (see Lemma 4), we acquire the following result.

**Theorem 6:** For $n \geq 3$, the balanced hypercube $BH_n$ admits two CISTS with diameter $6n - 2$.

**Acknowledgments**

This research was partially supported by MOST grants 107-2221-E-131-001 (Kung-Jui Pai), 107-2221-E-141-002 (Ruay-Shiuang Chang) and 107-2221-E-141-001-MY3 (Jou-Ming Chang), from the Ministry of Science and Technology, Taiwan.

**References**