Independent Spanning Trees on Crossed Cubes*

Jhen-Ding Wang[1] Jou-Ming Chang[1,†] Jinn-Shyong Yang[2] Kun-Fu Ding[3]

[1] Institute of Information and Decision Sciences,
National Taipei College of Business, Taipei, Taiwan, ROC

[2] Department of Information Management,
National Taipei College of Business, Taipei, Taiwan, ROC

[3] Department of Industrial Engineering and Management,
Ming Chi University of Technology, New Taipei City, Taiwan, ROC

Abstract

A set of spanning trees in a graph is said to be independent (ISTs for short) if all the trees are rooted at the same node r and for any other node $v(\neq r)$, the paths from v to r in any two trees are node-disjoint except the two end nodes v and r. For an n-connected graph, the independent spanning trees problem asks to construct n ISTs rooted at an arbitrary node of the graph. Recently, Zhang et al. [Y.-H. Zhang, W. Hao, and T. Xiang, Independent spanning trees in crossed cubes, Inform. Process. Lett., 113 (2013) 653–658] proposed an algorithm to construct n ISTs with a common root at node 0 in an n-dimensional crossed cube CQ_n. However, it has been proved by Kulasinghe and Bettayeb [P.D. Kulasinghe and S. Bettayeb, Multiply-twisted hypercube with 5 or more dimensions is not vertex transitive, Inform. Process. Lett., 53 (1995) 33–36] that the CQ_n (a synonym called multiply-twisted hypercube in that paper) fails to be node-transitive for $n \geqslant 5$. Thus, the result of Zhang et al. does not really solve the ISTs problem in CQ_n. In this paper, we revisit the problem of constructing n ISTs rooted at an arbitrary node in CQ_n. As a consequence, we show that the proposed algorithm can be parallelized to run in $\mathcal{O}(\log N)$ time using $N = 2^n$ nodes of CQ_n as processors.

Keyword: independent spanning trees; interconnection networks; crossed cubes; multiply-twisted hypercube;

1 Introduction

Constructing multiple spanning trees in networks have been studied from not only the theoretical point of view but also some practical applications such as fault-tolerant broadcasting [1, 15] and secure message distribution [1, 25, 31]. Let G be a graph with node set $V(G)$ and edge set $E(G)$, respectively. Two spanning trees in a graph G are said to be *independent* if they are rooted at the same node r such that, for each node $v(\neq r)$ in G, the two different paths from v to r, one path in each tree, are internally node-disjoint. A set of spanning trees of G is called *independent spanning trees* (ISTs for short) if they are pairwise independent.

A graph G is k-*connected* if $|V(G)| > k$ and $G-F$ is connected for every subset $F \subseteq V(G)$ with $|F| < k$, where $G-F$ denotes the graph obtained from G by removing F. It was conjectured by Zehavi and Itai [38] that for any n-connected graph there exist n ISTs rooted at an arbitrary node. From then on, this conjecture has been shown to be true for k-connected graphs with $k \leqslant 4$ (see [15], [8,38] and [9] for $k = 2, 3, 4$, respectively) and is still open for $k \geqslant 5$. In particular, this conjecture has been confirmed for several restricted classes of graphs, e.g., graphs related to planarity [13,14,22,23], graphs defined by Cartesian product [3, 24, 26, 27, 30, 33, 37], variations of hypercubes [4–7, 21, 28, 29, 31], special Cayley graphs [17, 18, 25, 32, 35, 36], and chordal ring [16, 34].

The n-dimensional crossed cube CQ_n, proposed first by Efe [11], is a variant of an n-dimensional hypercube. One advantage of CQ_n is that the diameter is only about one half of the diameter of an n-dimensional hypercube. For more properties

*This research was partially supported by National Science Council under the Grants NSC102-2221-E-141-002 and NSC102-2221-E-141-001-MY3.

†Corresponding author. Email: spade@mail.ntcb.edu.tw

of CQ_n, the reader can refer to [2, 10, 12, 19, 20]. Note that Kulasinghe [19] showed that CQ_n is n-connected. Cheng et al. [6] and [5] respectively proposed algorithms to construct n ISTs rooted at an arbitrary node in CQ_n. Let $N = 2^n$. The construction scheme of [6] is in a recursive fashion to run in $\mathcal{O}(N \log^2 N)$ time. Although the algorithm in [5] can simultaneously construct n ISTs in parallel with time complexity $\mathcal{O}(N)$, it is not fully parallelized for the construction of each spanning tree. Recently, Zhang et al. [39] proposed another algorithm that takes time $\mathcal{O}(N \log N)$ for constructing n ISTs rooted at node 0 in CQ_n and showed that it can be parallelized to run in time $\mathcal{O}(\log N)$. Because Kulasinghe and Bettayeb [20] had already pointed out that CQ_n (a synonym called multiply-twisted hypercube in that paper) fails to be node-transitive for $n \geqslant 5$, the construction of [39] that takes node 0 as the common root of spanning trees does not really solve the ISTs problem in CQ_n. In this paper, we present a fully parallelized approach for constructing n ISTs rooted at an arbitrary node in CQ_n. Our algorithm totally takes $\mathcal{O}(N \log N)$ time and can be parallelized to run in $\mathcal{O}(\log N)$ time using $N = 2^n$ nodes of CQ_n as processors.

The rest of this paper is organized as follows. Section 2 formally gives the definition of crossed cubes and provides some useful terminologies and notations. Section 3 presents our algorithm for constructing ISTs in CQ_n. The final section proves the correctness of the algorithm.

2 Preliminary

In this paper, we use a binary string $x_{n-1}x_{n-2}\cdots x_1x_0$ of length n to label a node x in CQ_n. Two binary strings $x = x_1x_0$ and $y = y_1y_0$ are *pair-related*, denoted $x \sim y$, if and only if $(x, y) \in \{(00, 00), (10, 10), (01, 11), (11, 01)\}$. The n-dimensional crossed cube CQ_n is the labeled graph with the following recursively fashion:

CQ_1 is the complete graph on two nodes with labels 0 and 1. For $n \geqslant 2$, CQ_n consists of two subcubes CQ^0_{n-1} and CQ^1_{n-1} such that every vertex in CQ^0_{n-1} and CQ^1_{n-1} is labeled by 0 and 1 in its leftmost bit, respectively. Two nodes $x = 0x_{n-2}\cdots x_1x_0 \in V(CQ^0_{n-1})$ and $y = 1y_{n-2}\cdots y_1y_0 \in V(CQ^1_{n-1})$ are joined by an edge if and only if

(1) $x_{n-2} = y_{n-2}$ if n is even, and
(2) $x_{2i+1}x_{2i} \sim y_{2i+1}y_{2i}$ for $0 \leqslant i < \lfloor (n-1)/2 \rfloor$.

Figure 1 shows crossed cubes CQ_3 and CQ_4.

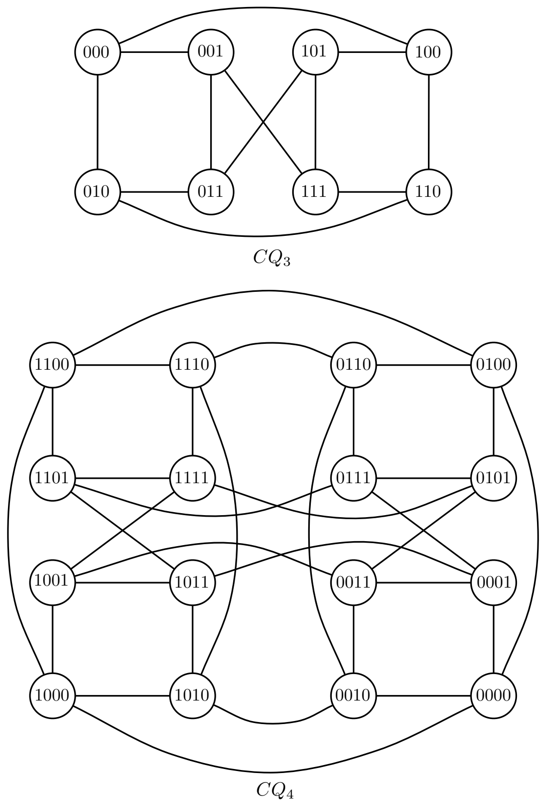

Figure 1: Croosed cubes CQ_3 and CQ_4.

Let $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$. Crossed cubes can be defined equivalently as follows:

Lemma 1. [11] *For all integer $n \geqslant 1$, two nodes $x = x_{n-1}x_{n-2}\cdots x_0$ and $y = y_{n-1}y_{n-2}\cdots y_0$ are joined by an edge in CQ_n if and only if there exists an integer $i \in \mathbb{Z}_n$ such that*

(1) $x_{n-1}x_{n-2}\cdots x_{i+1} = y_{n-1}y_{n-2}\cdots y_{i+1}$,

(2) $x_i \neq y_i$,

(3) $x_{i-1} = y_{i-1}$ *if i is odd, and*

(4) $x_{2j+1}x_{2j} \sim y_{2j+1}y_{2j}$ *for* $0 \leqslant j < \lfloor i/2 \rfloor$.

If conditions (1) and (2) of Lemma 1 hold, we say that x and y have the *leftmost differing bit at position i*. In this case, x and y are said to be the *i-neighbors* to each other, and for notational convenience we write $y = N_i(x)$ or $x = N_i(y)$. Moreover, the edge (x, y) is an *i-dimensional edge* of CQ_n, and we denote $i = \dim(x, y)$. For example, we consider the node $x = 011011$ in CQ_6. Then, $N_i(x)$ for $i = 0, 1, \ldots, 5$ are 011010, 011001, 011101, 010001, 001001, and 111001, respectively.

In this paper, we also use the following notation. Two paths P and Q joining two distinct nodes x and y are *internally node-disjoint*, denoted by $P||Q$, if $V(P) \cap V(Q) = \{x, y\}$. Let T be a spanning tree rooted at node r of CQ_n. The parent of a node $x(\neq r)$ in T is denoted by $\text{PARENT}(T, x)$. For $x, y \in V(T)$, the unique path from x to y is denoted by $T[x, y]$. Hence, two spanning trees T and T' with the same root r are ISTs if and only if $T[x, r] \,||\, T'[x, r]$ for every node $x \in V(T) \setminus \{r\}$.

3 An algorithm of Constructing ISTs

Since CQ_n is n-connected and we would like to construct n ISTs, the root in each spanning tree must have a unique child. Let $r = r_{n-1}r_{n-2}\cdots r_0$ be the common root of ISTs. For $i \in \mathbb{Z}_n$, we denote T_i as a tree such that r takes its i-neighbor as the unique child. Let $N_i(r) = c_{n-1}c_{n-2}\cdots c_0$. A node is called the *surrenal* of $N_i(r)$, denoted by $\bar{N}_i(r) = c'_{n-1}c'_{n-2}\cdots c'_0$, if the following conditions hold:

(1) $c_j = c'_j$ for $j \geqslant i$ if i is even,
(2) $c_j = c'_j$ for $j > i$ if i is odd, and
(3) $c_{2j+1}c_{2j} \sim c'_{2j+1}c'_{2j}$ for $0 \leqslant j < \lceil i/2 \rceil$.

For each node $x = x_{n-1}x_{n-2}\cdots x_0 \in V(T_i) \setminus \{r\}$, a node $x' = x'_{n-1}x'_{n-2}\cdots x'_0$ with respect to x is defined as follows: $x_{2j+1}x_{2j} \sim x'_{2j+1}x'_{2j}$ for $0 \leqslant j < \lfloor n/2 \rfloor$ and $x_{n-1} = x'_{n-1}$ when n is odd. Let $I_i(x) = \{j \in \mathbb{Z}_n\colon x_j \neq c_j \text{ and } j > i\}$ and $I_i(x') = \{j \in \mathbb{Z}_n\colon x'_j \neq c_j \text{ and } j > i\}$. For two set of integers S and T, define the following function:

$$\beta(S,T) = \begin{cases} 0 & \text{if } S = \emptyset; \\ \beta(\{t \in T\colon t < \max S\}, S) + 1 & \text{otherwise.} \end{cases}$$

In particular, we let $\alpha_i(x) = \beta(I_i(x), I_i(x'))$. According to the parity of $\alpha_i(x)$, let

$$H_i(x) = \begin{cases} \{j \in \mathbb{Z}_n\colon x_j \neq c_j\} & \text{if } \alpha_i(x) \text{ is even;} \\ \{j \in \mathbb{Z}_n\colon x_j \neq c'_j\} & \text{otherwise.} \end{cases} \quad (1)$$

We further define the following function: $\textsc{next}(i,x) =$

$$\begin{cases} i & \text{if } H_i(x) = \emptyset; \\ \max H_i(x) & \text{if } H_i(x) \neq \emptyset \text{ and } i < \min H_i(x); \\ \max\{j \in H_i(x)\colon j \leqslant i\} & \text{otherwise.} \end{cases} \quad (2)$$

That is, we regard $H_i(x)$ as a cyclic ordered set in decreasing order. If $H_i(x) = \emptyset$ or $i \in H_i(x)$, the function outputs i; otherwise, the function outputs the next element in the cyclic order of $H_i(x)$ with respect to i.

For example, consider CQ_{12} and a node $x = 110001101110$ in T_4 rooted at $r = 101101000111$. By definitions, $N_4(r) = 101101011101$, $\bar{N}_4(r) = 101101010111$ and $x' = 010011100110$. Since $I_4(x) = \{10, 9, 8, 5\}$ and $I_4(x') = \{11, 10, 9, 8, 7, 5\}$, we can find $\alpha_4(x)$ as follows:

$$\begin{aligned} \alpha_4(x) &= \beta(\{10,9,8,5\}, \{11,10,9,8,7,5\}) \\ &= \beta(\{9,8,7,5\}, \{10,9,8,5\}) + 1 \\ &= \beta(\{8,5\}, \{9,8,7,5\}) + 2 \\ &= \beta(\{7,5\}, \{8,5\}) + 3 \\ &= \beta(\{5\}, \{7,5\}) + 4 \\ &= \beta(\emptyset, \{5\}) + 5 \\ &= 5. \end{aligned}$$

Thus, $H_4(x) = \{10, 9, 8, 5, 4, 3, 0\}$ and $\textsc{next}(4, x) = 4$. Table 1 shows more examples of CQ_6.

It is clear that, for each node $x \in V(CQ_n) \setminus \{r\}$, finding $I_i(x)$, $I_i(x')$, $\alpha_i(x)$, $H_i(x)$ and $\textsc{next}(i, x)$ can be done in $\mathcal{O}(n)$ time provided i is given. In what follows, we present a fully parallelized algorithm for constructing n spanning trees with an arbitrary node $r = r_{n-1}r_{n-2}\cdots r_0$ as their common root in CQ_n. For each node $x \in V(CQ_n) \setminus \{r\}$ with binary string $x = x_{n-1}x_{n-2}\cdots x_0$, the construction can be carried out by describing the parent of x in each spanning tree T_i.

Algorithm Constructing-ISTs
Input: All nodes of CQ_n and the common root $r = r_{n-1}r_{n-2}\cdots r_0$.
Output: n ISTs $T_0, T_1, \ldots, T_{n-1}$ root at r.

```
for i = 0 to n − 1 do in parallel
        /* construct T_i simultaneously */
    for each node x in CQ_n do in parallel
        /* generate parent of each node x
        simultaneously */
        j = NEXT(i, x)
        PARENT(T_i, x) = N_j(x)
```

Figure 2: Algorithm for constructing n spanning trees in CQ_n.

Table 1: The parent of some nodes $x \in V(CQ_6)$ in T_2 rooted at $r = 011011_2 = 27$.

$N_2(27) = 011101_2 = 29$
$\bar{N}_2(27) = 011111_2 = 31$

x	x'	$I_2(x)$	$I_2(x')$	$\alpha_2(x)$	$H_2(x)$	$j = \textsc{next}(2,x)$	$\textsc{parent}(T_2, x)$
(34) 100010	100010	$\{5,4,3\}$	$\{5,4,3\}$	3	$\{5,4,3,2,0\}$	2	$= N_2(34) = 100110$ (38)
(38) 100110	101110	$\{5,4,3\}$	$\{5,4\}$	3	$\{5,4,3,0\}$	0	$= N_0(38) = 100111$ (39)
(39) 100111	101101	$\{5,4,3\}$	$\{5,4\}$	3	$\{5,4,3\}$	5	$= N_5(39) = 001101$ (13)
(13) 001101	000111	$\{4\}$	$\{4,3\}$	2	$\{4\}$	4	$= N_4(13) = 010111$ (23)
(23) 010111	111101	$\{3\}$	$\{5\}$	1	$\{3\}$	3	$= N_3(23) = 011101$ (29)
(29) 011101	110111	$\emptyset$	$\{5,3\}$	0	$\emptyset$	2	$= N_2(29) = 011011$ (27)

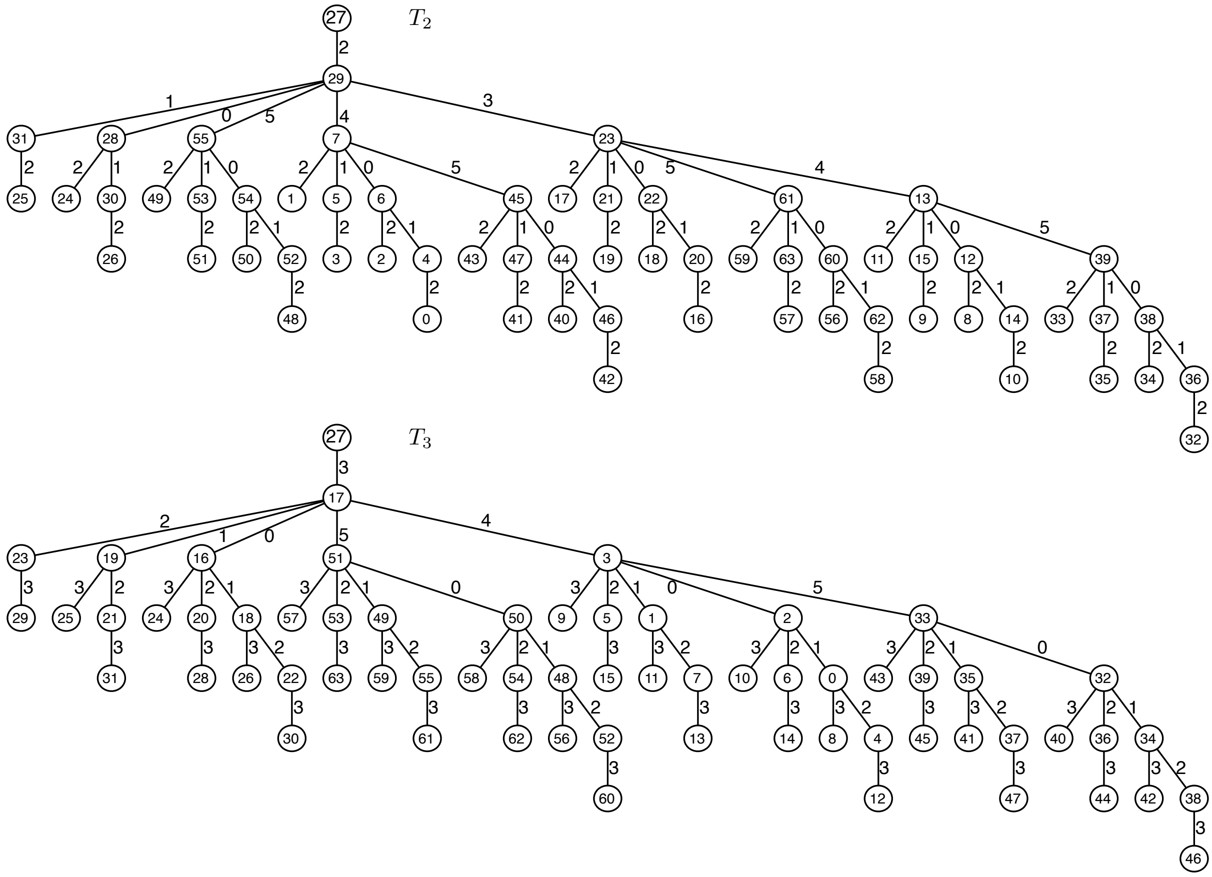

Figure 3: Two ISTs T_2 and T_3 of CQ_6.

Figure 3 illustrates the construction of T_2 and T_3 for CQ_6. Henceforth, we adopt the notation $x \xrightarrow{j} y$ to mean that $y = \text{PARENT}(T_i, x) = N_j(x)$ in T_i. For instance, we have $T_2[34, 27] = 34 \xrightarrow{2} 38 \xrightarrow{0} 39 \xrightarrow{5} 13 \xrightarrow{4} 23 \xrightarrow{3} 29 \xrightarrow{2} 27$ in Figure 3.

4 Correctness and analysis

In this section, we will show the validity of the algorithm. Firstly, we give the following basic property.

Lemma 2. *For $i \in \mathbb{Z}_n$ and a node $x \in V(CQ_n) \setminus \{r\}$, if $H_i(x) = \emptyset$ then $x = N_i(r)$.*

Proof. Suppose $H_i(x) = \emptyset$. We claim $\alpha_i(x) = 0$, and thus by Eq. (1), it follows that $x = N_i(r)$. We suppose that, on the contrary, $\alpha_i(x) \neq 0$ (i.e, $I_i(x) \neq \emptyset$). This implies that there is a $k \in \mathbb{Z}_n \setminus \mathbb{Z}_i$ such that $x_k \neq c_k$. Obviously, if $\alpha_i(x)$ is even, then $I_i(x) \subseteq H_i(x)$. This contradicts that $H_i(x) = \emptyset$. On the other hand, from the surrenal of $N_i(r)$, we have $c_j = c'_j$ for all $j > i$. Thus, $x_k \neq c'_k$, and it follows that $H_i(x) \neq \emptyset$, a contradiction. □

For two ordered sets A and B, we write $A \prec_{\text{LEX}} B$ to mean that A precedes B in lexicographic order. We now prove the reachability between every node $x(\neq r)$ and the root r in T_i, thereby proving the existence of a unique path from x to the root in the tree.

Theorem 3. *Let $r \in V(CQ_n)$ be an arbitrary node. The construction of T_i for $i \in \mathbb{Z}_n$ are spanning trees rooted at r.*

Proof. From Constructing-ISTs, since every node $v \in V(CQ_n)$ must be contained in T_i, it follows that T_i is a spanning subgraph of CQ_n. Let $x = x_{n-1}x_{n-2}\cdots x_0$ be any node of CQ_n. We show that $T_i[x, r]$ is the unique path connecting x and r in T_i. By Lemma 2, if $H_i(x) = \emptyset$, then $x = N_i(r)$. Thus, $\text{NEXT}(i, x) = i$ and $T_i[x, r] = x \xrightarrow{i} r$ is the desired path that connects x and r in T_i.

Next, we suppose that $H_i(x) = \{j_{p-1}, j_{p-2}, \ldots, j_0\}$ is nonempty and it is treated as an ordered set such that $j_{p-1} > j_{p-2} > \cdots > j_0$. Clearly, $1 \leqslant p \leqslant n$. There are two scenarios as follows:

Case 1: $i \notin H_i(x)$ (i.e., $x_i = c_i$). Let $j_k = \text{NEXT}(i, x)$, where $0 \leqslant k \leqslant p-1$. By Eq. (2), we know that $j_{p-1} > j_{p-2} > \cdots > j_{k+1} > i > j_k > \cdots > j_0$. Since $H_i(x) \neq \emptyset$, we assume that $y(\neq r) = y_{n-1}y_{n-2}\cdots y_0$ is the parent of x in T_i. That is, $y = \text{PARENT}(T_i, x) = N_{j_k}(x)$. By Lemma 1, the following condition hold: (i) $y_{n-1}y_{n-2}\cdots y_{j_k+1}y_{j_k} = x_{n-1}x_{n-2}\cdots x_{j_k+1}\bar{x}_{j_k}$; (ii) $y_{j_k-1} = x_{j_k-1}$ when j_k is odd; and (iii) $y_{2j+1}y_{2j} \sim x_{2j+1}x_{2j}$ for $0 \leqslant j < \lfloor j_k/2 \rfloor$. We consider the following two subcases:

Case 1.1: $\alpha_i(x)$ is even. By Eq. (1), $x_j \neq c_j$ for $j \in H_i(x)$ and $x_j = c_j$ for $j \notin \mathbb{Z}_n \setminus H_i(x)$. Thus, we have $I_i(x) = H_i(x) \setminus \{j_k, j_{k-1}, \cdots, j_0\}$. Since $i > j_k$, we have $y_j = x_j$ for every bit at position j with $j > i$. Thus, $I_i(y) = I_i(x)$. In addition, for $j_k < j \leqslant i$, we have $y_j = x_j = c_j$. Moreover, $x_{j_k} \neq c_{j_k}$ and $y_{j_k} \neq x_{j_k}$ imply $y_{j_k} = c_{j_k}$. Let $F = \{j \in \mathbb{Z}_{j_k}\colon y_j \neq c_j\}$. Then, we can determine $H_i(y)$ as follows: $H_i(y) = I_i(y) \cup F = (H_i(x) \setminus \{j_k, j_{k-1}, \cdots, j_0\}) \cup F$.

Case 1.2: $\alpha_i(x)$ is odd. By Eq. (1), $x_j \neq c'_j$ for $j \in H_i(x)$ and $x_j = c'_j$ for $j \notin \mathbb{Z}_n \setminus H_i(x)$. Let $I'_i(x) = \{j \in \mathbb{Z}_n\colon x_j \neq c'_j \text{ and } j > i\}$. Clearly, $I'_i(x) = H_i(x) \setminus \{j_k, j_{k-1}, \cdots, j_0\}$. Since $i > j_k$, we have $y_j = x_j$ for every bit at position j with $j > i$. Thus, $I'_i(y) = I'_i(x)$. In addition, for $j_k < j \leqslant i$, we have $y_j = x_j = c'_j$. Moreover, $x_{j_k} \neq c'_{j_k}$ and $y_{j_k} \neq x_{j_k}$ imply $y_{j_k} = c'_{j_k}$. Let $F = \{j \in \mathbb{Z}_{j_k}\colon y_j \neq c'_j\}$. Then, we can determine $H_i(y)$ as follows: $H_i(y) = I'_i(y) \cup F = (H_i(x) \setminus \{j_k, j_{k-1}, \cdots, j_0\}) \cup F$.

From above, we can determine $H_i(y)$. In particular, we show that $H_i(y) \prec_{\text{LEX}} H_i(x)$ and $j_k \notin H_i(y)$. By a similar argument, if $H_i(y) \neq \emptyset$, let $z = \text{PARENT}(T_i, y) = N_{j_\ell}(y)$ be the parent of y in T_i, where $j_\ell = \text{NEXT}(i, y)$. Again, we can determine $H_i(z)$ and show that $j_k, j_\ell \notin H_i(z)$. By this way, we find a sequence of nodes $y, z, \cdots, c = N_i(r)$ in T_i such that $H_i(c) = \emptyset$. Recall that we have already constructed $T_i[c, r] = c \xrightarrow{i} r$ for connecting c and r in T_i before Case 1. Therefore, we obtain the following unique path that connects x and r in T_i:

$$T_i[x, r] : x \xrightarrow{j_k} y \xrightarrow{j_\ell} z \xrightarrow{j_m} \cdots \xrightarrow{j_q} c \xrightarrow{i} r.$$

Case 2: $i \in H_i(x)$ (i.e., $x_i \neq c_i$). Suppose $i = j_k$ for some $k \in \{0, 1, \ldots, p-1\}$. By Eq. (2), we have $\text{NEXT}(i, x) = i$. Let $y = \text{PARENT}(T_i, x) = N_i(x)$. Clearly, $y_i = \bar{x}_i = c_i$. This shows that the current status of y is in the situation of Case 1. Let $P = T_i[y, r]$ be the path connecting y and r in T_i. Therefore, we obtain the unique path $T_i[x, r]$ by concatenating $x \xrightarrow{i} y$ and P. □

According to the proof of Theorem 3, we have the following properties.

Corollary 4. *For $i \in \mathbb{Z}_n$, let $T_i[x, r] : v_0(= x) \xrightarrow{j_1} v_1 \xrightarrow{j_2} \cdots \xrightarrow{j_k} v_k \xrightarrow{i} r$ be a path constructed from Theorem 3. Then, the following statements hold:*

(1) $\emptyset = H_i(v_k) \prec_{\text{LEX}} H_i(v_{k-1}) \prec_{\text{LEX}} \cdots \prec_{\text{LEX}} H_i(v_0)$.

(2) *For $1 \leqslant \ell < m \leqslant k$, $j_\ell \notin H_i(v_m)$ (i.e., $j_\ell \neq j_m$).*

(3) *For $2 \leqslant \ell \leqslant k$, $j_\ell \neq i$. In particular, it is possible $j_1 = i$.*

For instance, if we consider the path $T_2[34, 27] = 34 \xrightarrow{2} 38 \xrightarrow{0} 39 \xrightarrow{5} 13 \xrightarrow{4} 23 \xrightarrow{3} 29 \xrightarrow{2} 27$ in Figure 3, we can verify from Table 1 as follows: $(H_2(29) = \emptyset) \prec_{\text{LEX}} (H_2(23) = \{3\}) \prec_{\text{LEX}} (H_2(13) = \{4\}) \prec_{\text{LEX}} (H_2(39) = \{5, 4, 3\}) \prec_{\text{LEX}} (H_2(38) = \{5, 4, 3, 0\}) \prec_{\text{LEX}} (H_2(34) = \{5, 4, 3, 2, 0\})$. Let $\text{HEIGHT}(T)$ denote the height of a tree T. Since $|H_i(x)| \leqslant n$ for every node $x \in V(CQ_n)$, the following result can be obtained from Corollary 4 directly.

Corollary 5. *For $i \in \mathbb{Z}_n$, $\text{HEIGHT}(T_i) \leqslant n + 1$.*

Theorem 6. *The spanning trees constructed from* CONSTRUCTING-ISTS *are independent.*

Proof. We prove the lemma by contradiction. Suppose that the lemma is false. That is, there exist two integers $i, j \in \mathbb{Z}_n$ and a node $x \in V(CQ_n) \setminus \{r\}$ such that the following two paths constructed in Theorem 3 satisfy $\{x, r\} \subsetneq P \cap Q$:

$$P = T_i[x, r] : u_0(= x) \xrightarrow{j_0} u_1 \xrightarrow{j_1} u_2 \xrightarrow{j_2} \cdots \xrightarrow{j_{k-1}} u_k \xrightarrow{i} r$$

and

$$Q = T_j[x, r] : v_0(= x) \xrightarrow{\ell_0} v_1 \xrightarrow{\ell_1} v_2 \xrightarrow{\ell_2} \cdots \xrightarrow{\ell_{m-1}} v_m \xrightarrow{j} r.$$

Suppose that $u_p = v_q$ for $1 \leqslant p < k$ and $1 \leqslant q < m$. Let $A = \{j_p, j_{p+1}, \ldots, j_{k-1}, i\}$ and $B = \{\ell_q, \ell_{q+1}, \ldots, \ell_{m-1}, j\}$. Since $i \neq j$, by Corollary 4 we have $A \neq B$. Let $d = \max((A \cup B) \setminus (A \cap B))$. This implies that the dth bit of u_p is different from that of v_q, which leads to a contradiction. □

According to Theorems 3 and 6, we have the following main result.

Corollary 7. *Let $N = 2^n$ and $r \in V(CQ_n)$ be an arbitrary node. Algorithm* CONSTRUCTING-ISTS *can correctly construct n ISTs rooted at r in $\mathcal{O}(N \log N)$ time. In particular, the algorithm can be parallelized to run in $\mathcal{O}(\log N)$ time using N processors of CQ_n.*

References

[1] F. Bao, Y. Funyu, Y. Hamada and Y. Igarashi, Reliable broadcasting and secure distributing

in channel networks, in: *Proc. of 3rd International Symposium on Parallel Architectures, Algorithms and Networks*, ISPAN'97, Taipei, December 1997, pp. 472–478.

[2] C.-P. Chang, T.-Y. Sung, and L.-H. Hsu, Edge congestion and topological properties of crossed cube, *IEEE Trans. Parallel Distrib. Syst.*, 11 (2000) 64–80

[3] X.-B. Chen, Parallel construction of optimal independent spanning trees on Cartesian product of complete graphs, *Inform. Process. Lett.*, 111 (2011) 235–238.

[4] B. Cheng, J. Fan, X. Jia, and J. Jia, Parallel construction of independent spanning trees and an application in diagnosis on Möbius cubes, *J. Supercomput.*, 65 (2013) 1279–1301.

[5] B. Cheng, J. Fan, X. Jia, and J. Wang, Dimension-adjacent trees and parallel construction of independent spanning trees on crossed cubes, *J. Parallel Distrib. Comput.*, 73 (2013) 641–652.

[6] B. Cheng, J. Fan, X. Jia, and S. Zhang, Independent spanning trees in crossed cubes, *Inform. Sci.*, 233 (2013) 276–289.

[7] B. Cheng, J. Fan, X. Jia, S. Zhang, and B. Chen, Constructive algorithm of independent spanning trees on Möbius cubes, *Comput. J.*, 56 (2013) 1347–1362.

[8] J. Cheriyan and S.N. Maheshwari, Finding nonseparating induced cycles and independent spanning trees in 3-connected graphs, *J. Algorithms*, 9 (1988) 507–537.

[9] S. Curran, O. Lee and X. Yu, Finding four independent trees, *SIAM J. Comput.*, 35 (2006) 1023–1058.

[10] K. Efe, A variation on the hypercube with lower diameter, *IEEE Trans. Comput.*, 40 (1991) 1312–1316.

[11] K. Efe, The crossed cube architecture for parallel computation, *IEEE Trans. Parallel Distrib. Syst.*, 3 (1992) 513–524.

[12] K. Efe, P.K. Blackwell, W. Slough, T. Shiau, Topological properties of the crossed cube architecture, *Parallel Comput.*, 20 (1994) 1763–1775.

[13] A. Huck, Independent trees in graphs, *Graphs Combin.*, 10 (1994) 29–45.

[14] A. Huck, Independent trees in planar graphs, *Graphs Combin.*, 15 (1999) 29–77.

[15] A. Itai and M. Rodeh, The multi-tree approach to reliability in distributed networks, *Inform. Comput.*, 79 (1988) 43–59.

[16] Y. Iwasaki, Y. Kajiwara, K. Obokata, and Y. Igarashi, Independent spanning trees of chordal rings, *Inform. Process. Lett.*, 69 (1999) 155–160.

[17] J.-S. Kim, H.-O. Lee, E. Cheng, and L. Lipták, Optimal independent spanning trees on odd graphs, *J. Supercomputing*, 56 (2011) 212–225.

[18] J.-S. Kim, H.-O. Lee, E. Cheng, and L. Lipták, Independent spanning trees on even networks, *Inform. Sci.*, 181 (2011) 2892–2905.

[19] P.D. Kulasinghe, Connectivity of the crossed cube, *Inform. Process. Lett.*, 61 (1997) 221–226.

[20] P.D. Kulasinghe and S. Bettayeb, Multiply-twisted hypercube with 5 or more dimensions is not vertex transitive, *Inform. Process. Lett.*, 53 (1995) 33–36.

[21] Y-J. Liu, J.K. Lan, W.Y. Chou, and C. Chen, Constructing independent spanning trees for locally twisted cubes, *Theoret. Comput. Sci.*, 412 (2011) 2237–2252.

[22] K. Miura, S. Nakano, T. Nishizeki, and D. Takahashi, A linear-time algorithm to find four independent spanning trees in four connected planar graphs, *Internat. J. Found. Comput. Sci.*, 10 (1999) 195–210.

[23] S. Nagai and S. Nakano, A linear-time algorithm to find independent spanning trees in maximal planar graphs, *IEICE Trans. Fund. Electron. Comm. Comput. Sci.*, E84-A (2001) 1102–1109.

[24] K. Obokata, Y. Iwasaki, F. Bao, and Y. Igarashi, Independent spanning trees of product graphs and their construction, *IEICE Trans. Fund. Electron. Comm. Comput. Sci.*, E79-A (1996) 1894–1903.

[25] A.A. Rescigno, Vertex-disjoint spanning trees of the star network with applications to fault-tolerance and security, *Inform. Sci.*, 137 (2001) 259–276.

[26] S.-M. Tang, Y.-L. Wang, and Y.-H. Leu, Optimal independent spanning trees on hypercubes, *J. Inform. Sci. Eng.*, 20 (2004) 143–155.

[27] S.-M. Tang, J.-S. Yang, Y.-L. Wang, and J.-M. Chang, Independent spanning trees on mul-

tidimensional torus networks, *IEEE Trans. Comput.*, 59 (2010) 93–102.

[28] Y. Wang, J. Fan, G. Zhou, and X. Jia, Independent spanning trees on twisted cubes, *J. Parallel Distrib. Comput.*, 72 (2012) 58–69.

[29] Y. Wang, J. Fan, X. Jia, and H. Huang, An algorithm to construct independent spanning trees on parity cubes, *Theoret. Comput. Sci.*, 465 (2012) 61–72.

[30] J. Werapun, S. Intakosum, and V. Boonjing, An efficient parallel construction of optimal independent spanning trees on hypercubes, *J. Parallel Distrib. Comput.*, 72 (2012) 1713–1724.

[31] J.-S. Yang, H.-C. Chan, and J.-M. Chang, Broadcasting secure messages via optimal independent spanning trees in folded hypercubes, *Discrete Appl. Math.*, 159 (2011) 1254–1263.

[32] J.-S. Yang and J.-M. Chang, Independent spanning trees on folded hyper-stars, *Networks*, 56 (2010) 272–281.

[33] J.-S. Yang and J.-M. Chang, Optimal independent spanning trees on Cartesian product of hybrid graphs, *Comput. J.*, 57 (2014) 93–99.

[34] J.-S. Yang, J.-M. Chang, S.-M. Tang, and Y.-L. Wang, Reducing the height of independent spanning trees in chordal rings, *IEEE Trans. Parallel Distrib. Syst.*, 18 (2007) 644–657.

[35] J.-S. Yang, J.-M. Chang, S.-M. Tang, and Y.-L. Wang, On the independent spanning trees of recursive circulant graphs $G(cd^m, d)$ with $d > 2$, *Theoret. Comput. Sci.*, 410 (2009) 2001–2010.

[36] J.-S. Yang, J.-M. Chang, S.-M. Tang, and Y.-L. Wang, Constructing multiple independent spanning trees on recursive circulant graphs $G(2^m, 2)$, *Int. J. Found. Comput. Sci.*, 21 (2010) 73–90.

[37] J.-S. Yang, S.-M. Tang, J.-M. Chang, and Y.-L. Wang, Parallel construction of optimal independent spanning trees on hypercubes, *Parallel Comput.*, 33 (2007) 73–79.

[38] A. Zehavi and A. Itai, Three tree-paths, *J. Graph Theory*, 13 (1989) 175–188.

[39] Y.-H. Zhang, W. Hao, and T. Xiang, Independent spanning trees in crossed cubes, *Inform. Process. Lett.*, 113 (2013) 653–658.