Theoretical Computer Science 412 (2011) 2237–2252

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Constructing independent spanning trees for locally twisted cubes[☆]

Yi-Jiun Liu, James K. Lan, Well Y. Chou, Chiuyuan Chen [*]

Department of Applied Mathematics, National Chiao Tung University, Hsinchu 300, Taiwan

ARTICLE INFO

Article history:
Received 11 December 2009
Accepted 27 December 2010
Communicated by D.-Z. Du

Keywords:
Independent spanning trees
Data broadcasting
Design and analysis of algorithms
Locally twisted cubes
Hypercubes
Hypercube variants
Parallel algorithm

ABSTRACT

The independent spanning trees (ISTs) problem attempts to construct a set of pairwise independent spanning trees and it has numerous applications in networks such as data broadcasting, scattering and reliable communication protocols. The well-known ISTs conjecture, Vertex/Edge Conjecture, states that any n-connected/n-edge-connected graph has n vertex-ISTs/edge-ISTs rooted at an arbitrary vertex r. It has been shown that the Vertex Conjecture implies the Edge Conjecture. In this paper, we consider the independent spanning trees problem on the n-dimensional locally twisted cube LTQ_n. The very recent algorithm proposed by Hsieh and Tu (2009) [12] is designed to construct n edge-ISTs rooted at vertex 0 for LTQ_n. However, we find out that LTQ_n is not vertex-transitive when $n \geq 4$; therefore Hsieh and Tu's result does not solve the Edge Conjecture for LTQ_n. In this paper, we propose an algorithm for constructing n vertex-ISTs for LTQ_n; consequently, we confirm the Vertex Conjecture (and hence also the Edge Conjecture) for LTQ_n.

1. Introduction

Two spanning trees in a graph G are said to be vertex/edge independent if they are rooted at the same vertex r and for each vertex v of G, $v \neq r$, the paths from r to v in two trees are vertex/edge disjoint except the two end vertices. A set of spanning trees of G are said to be vertex/edge independent if they are pairwise vertex/edge independent. The vertex/edge independent spanning trees (ISTs) problem attempts to construct a set of pairwise vertex/edge independent spanning trees and it has has applications such as data broadcasting, scattering and reliable communication protocols. For example, a rooted spanning tree in the underlying graph of a network can be viewed as a broadcasting scheme for data communication and fault-tolerance can be achieved by sending n copies of the message along the n independent spanning trees rooted at the source node [1]. For other applications, see [3] for the multi-node broadcasting problem, [21] for one-to-all broadcasting, and [2] for n-channel graphs, reliable broadcasting and secure message distribution.

The independent spanning trees problem has been widely studied in the last two decades. Two well-known conjectures on this problem are raised by Zehavi and Itai [27]: (refer to [4] or [23] for graph terminologies)

Conjecture 1.1 (*Vertex Conjecture*)**.** *Any n-connected graph has n vertex-ISTs rooted at an arbitrary vertex r.*

Conjecture 1.2 (*Edge Conjecture*)**.** *Any n-edge-connected graph has n edge-ISTs rooted at an arbitrary vertex r.*

Zehavi and Itai [27] also raised the question: It would be interesting to show that either the Vertex Conjecture implies the Edge Conjecture, or vice versa. Later, Khuller and Schieber [16] successfully proved that the Vertex Conjecture implies the Edge Conjecture, i.e., if any n-connected graph has n vertex-ISTs, then any n-edge-connected graph has n edge-ISTs. Khuller

☆ This research was partially supported by the National Science Council of the Republic of China under the grants grant NSC97-2628-M-009-006-MY3.
* Corresponding author. Tel.: +886 3 5731767.
E-mail addresses: cychen@mail.nctu.edu.tw, cychen@cc.nctu.edu.tw (C. Chen).

0304-3975/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2010.12.061

Table 1
The connectivity, edge-connectivity and diameters of Q_n and its variants.

Topology	$\kappa(G)$	$\lambda(G)$	Diameter	
Q_n	n	n	n	
LTQ_n	n	n	$\lceil (n+1)/2 \rceil$ $\lceil (n+3)/2 \rceil$	if $n < 5$ if $n \geq 5$
TQ_n	n	n	$\lceil (n+1)/2 \rceil$	
MQ_n	n	n	$\lceil (n+2)/2 \rceil$ $\lceil (n+1)/2 \rceil$	in 0-MQ_n for $n \geq 4$ in 1-MQ_n for $n \geq 1$

and Schieber's proof also works for the directed graphs. For the directed case, Edmonds [7] solved the Edge Conjecture. Khuller and Schieber [16] pointed out that the Vertex Conjecture for directed graphs is the strongest conjecture since it implies all the other conjectures.

The vertex and the edge conjectures have been confirmed only for $n \leq 4$. In particular, in [15], Itai and Rodeh proposed a linear-time algorithm for constructing two edge-ISTs for a 2-edge-connected graph; they also solved the Vertex Conjecture for $n = 2$. In [27], Zehavi and Itai solved the Vertex Conjecture for $n = 3$, but they did not proposed an algorithm for constructing three vertex-ISTs. In [6], Cheriyan and Maheshwari proposed an $O(|V(G)|^2)$-time algorithm for constructing three vertex-ISTs in a 3-connected graph. In [5], Curran et al. proposed an $O(|V(G)|^3)$-time algorithm for constructing four vertex-ISTs in a 4-connected graph. When $n \geq 5$, both the vertex and the edge conjectures are still open. It has been proven that the Vertex/Edge Conjecture holds for several restricted classes of graphs or digraphs, such as planar graphs [9,10,17,18], maximal planar graphs [19], product graphs [20], chordal rings [14,24], de Bruijn and Kautz digraphs [8,11], and hypercubes [22,26]. Note that the development of algorithms for constructing vertex-ISTs tends toward pursuing two research goals: One is to design efficient construction schemes (for example, [14,17,19,24] proposed linear-time algorithms) and the other is to reduce the heights of vertex-ISTs (for example, [11,22,24] proposed the idea of height improvements).

The hypercube (Q_n) is one of the most popular interconnection network topologies due to its simple structure and ease of implementation. Several commercial machines with hypercube topology have been built and a huge amount of research work, both theoretical and practical, has been done on various aspects of the hypercube. However, it has been shown that the hypercube does not achieve the smallest possible diameter for its resources. Therefore, many variants of the hypercube have been proposed. The most well-known variants are locally twisted cubes (LTQ_n), twisted cubes (TQ_n), crossed cubes (CQ_n) and Möbius cubes (MQ_n). A concise comparison including the connectivity, edge-connectivity and diameters of Q_n and its variants is shown in Table 1. Clearly, one advantage of LTQ_n over Q_n is that the diameter of LTQ_n is only about half of that of Q_n.

Before going further, we now briefly review results of the vertex-ISTs problem for Q_n. It is well known that Q_n is n-connected. Since Q_n is a product graph, the algorithm proposed by Obokata et al. [20] can be used to construct n vertex-ISTs for Q_n. As to the construction of the height-reduced vertex-ISTs on Q_n, Tang et al. [22] modified the algorithm in [20] and proposed an $O(n2^n)$-time algorithm for constructing an optimal set (in the sense of smallest average path lengths) of n vertex-ISTs for Q_n. It was pointed out by Yang et al. [26] that the algorithms in [20,22] are designed by a recursive fashion and such a construction forbids the possibility that the algorithm could be parallelized; Yang et al. [26] therefore proposed a parallel construction for an optimal set of n vertex-ISTs for Q_n.

The purpose of this paper is to confirm the Vertex Conjecture for the n-dimensional locally twisted cube LTQ_n. The very recent algorithm proposed by Hsieh and Tu [12] is designed to construct n edge-ISTs rooted at vertex 0 for LTQ_n. However, we find out that LTQ_n is not vertex-transitive whenever $n \geq 4$ (see Section 2). Therefore, Hsieh and Tu did not solve the Edge Conjecture for LTQ_n. In this paper, we will propose an algorithm for constructing n vertex-ISTs rooted at an arbitrary vertex of LTQ_n. Therefore, we will confirm the Vertex Conjecture for LTQ_n. Since vertex-ISTs are edge-ISTs, we also confirm the Edge Conjecture for LTQ_n.

In the remaining discussion, we will simply use ISTs to denote vertex-ISTs unless otherwise specified. This paper is organized as follows. In Section 2, we give definitions and notations used in the paper. In Section 3, we present an algorithm to construct n ISTs rooted at an arbitrary vertex of LTQ_n. In Section 4, we prove the correctness of our algorithm. Concluding remarks are given in the last section.

2. Preliminaries

All graphs in this paper are simple undirected graphs. Let G be a graph with vertex set $V(G)$ and the edge set $E(G)$. Let $x, y \in V(G)$. A path from x to y is denoted as x, y-path. The *distance* between two vertices x and y, denoted by $d(x, y)$, is the length of a shortest x, y-path. Two x, y-paths P and Q are *edge-disjoint* if $E(P) \cap E(Q) = \emptyset$. Two x, y-paths P and Q are *internally vertex-disjoint* if $V(P) \cap V(Q) = \{x, y\}$. A subgraph T of G is a *spanning tree* if T is a tree and $V(T) = V(G)$. Two spanning trees T and T' of G are *vertex-independent/edge-independent* if T and T' are rooted at the same vertex, say r, and

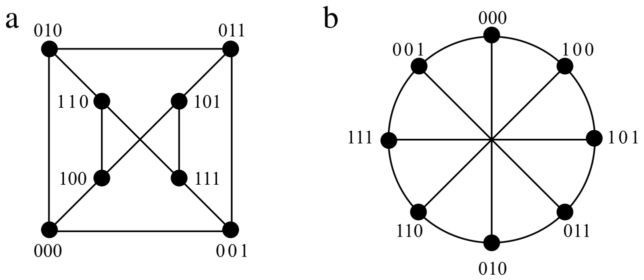

Fig. 1. (a) LTQ_3. (b) A symmetric drawing of LTQ_3.

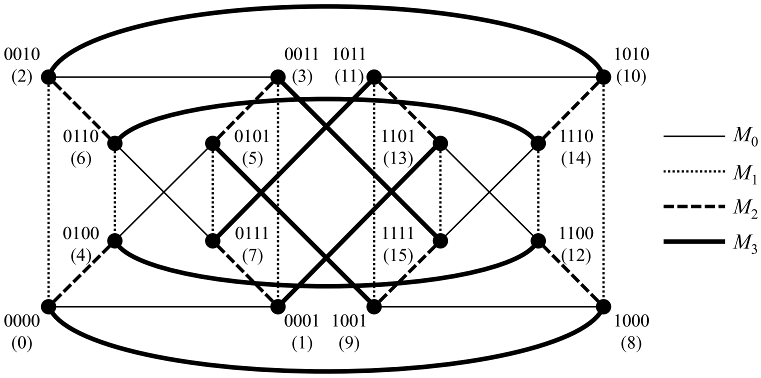

Fig. 2. LTQ_4 and its perfect matchings $\{M_0, M_1, M_2, M_3\}$.

for each $v \in V(G)$, $v \neq r$, the r, v-path in T and the r, v-path in T' are (internally) vertex-disjoint/edge-disjoint. A set of spanning trees of G are *vertex-independent/edge-independent* if they are pairwise vertex-independent/edge-independent.

2.1. The locally twisted cube

The *n-dimensional locally twisted cube* LTQ_n $(n \geq 2)$, proposed first by Yang et al. [25], has 2^n vertices. Each vertex is an n-string on $\{0, 1\}$, i.e., a binary string of length n. The LTQ_n is defined recursively as follows.

Definition 1 (*[25]*)**.** 1. LTQ_2 is the graph consisting of four vertices labeled with 00, 01, 10, and 11, respectively, and connected by the four edges (00, 01) (00, 10), (01, 11), and (10, 11).

2. LTQ_n $(n \geq 3)$ is built from two disjoint copies of LTQ_{n-1}'s as follows: Let $0LTQ_{n-1}$ (respectively, $1LTQ_{n-1}$) denote the graph obtained by prefixing the label of each vertex in one copy of LTQ_{n-1} with 0 (respectively, 1). Connect each vertex $0x_{n-2}x_{n-3}\dots x_0$ of $0LTQ_{n-1}$ to the vertex $1(x_{n-2} \oplus x_0)x_{n-3}\dots x_0$ of $1LTQ_{n-1}$ with an edge, where "$\oplus$" represents the XOR operation, or equivalently, the modulo 2 addition.

Figs. 1 and 2 illustrate LTQ_3 and LTQ_4, respectively. Yang et al. [25] also mentioned that the locally twisted cube can be equivalently defined by the following non-recursive fashion.

Definition 2 (*[25]*)**.** Let $x = x_{n-1}x_{n-2}\dots x_0$ and $y = y_{n-1}y_{n-2}\dots y_0$ be two vertices of LTQ_n $(n \geq 2)$. Then vertices x and y are adjacent if and only if one of the following conditions are satisfied.

1. There is an integer $2 \leq k \leq n-1$ such that
   (a) $x_k = \bar{y}_k$ ($\bar{y}_k$ is the complement of y_k in $\{0, 1\}$)
   (b) $x_{k-1} = y_{k-1} \oplus x_0$
   (c) all the remaining bits of x and y are identical.
2. There is an integer $0 \leq k \leq 1$ such that x and y only differ in the kth bit.

From Definition 2, LTQ_n is obviously an n-regular graph, and the labels of any two adjacent vertices of LTQ_n differ in at most two consecutive bits. Note that in the remaining part of this paper, the label of a vertex in LTQ_n is presented in binary representation and decimal representation interchangeably when there is no ambiguity.

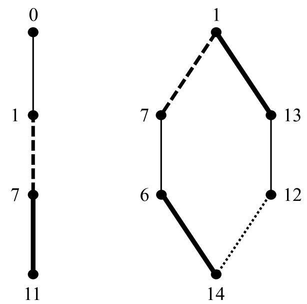

Fig. 3. The in-vertex-transitivity of LTQ_4.

2.2. The neighbor information and the perfect matchings of the locally twisted cube

From Definition 2, the n neighbors of an arbitrary vertex $x = x_{n-1}x_{n-2}\dots x_0$ of LTQ_n is given by

$$\begin{aligned}
f_0(x) &= x_{n-1}x_{n-2}x_{n-3}\dots x_2x_1\bar{x}_0,\\
f_1(x) &= x_{n-1}x_{n-2}x_{n-3}\dots x_2\bar{x}_1x_0,\\
f_2(x) &= x_{n-1}x_{n-2}x_{n-3}\dots \bar{x}_2\,(x_1\oplus x_0)\,x_0,\\
\vdots\; &= \qquad\qquad \vdots\\
f_{n-2}(x) &= x_{n-1}\bar{x}_{n-2}\,(x_{n-3}\oplus x_0)x_{n-4}\,\dots x_1x_0,\\
f_{n-1}(x) &= \bar{x}_{n-1}\,(x_{n-2}\oplus x_0)\,x_{n-3}\dots x_2x_1x_0,
\end{aligned} \tag{1}$$

where $f_k(x)$, $0 \le k \le n-1$, is called the *kth dimensional neighbor* of x; see also Lemma 4 in [13]. By (1), the n neighbors of vertices 0 and 1 can be determined as follows.

Lemma 2.1. *The n neighbors of vertex* 0 *in* LTQ_n *is given by*

$$f_k(0) = 2^k,$$

for $k = 0, 1, \dots, n-1$. *The n neighbors of vertex* 1 *in* LTQ_n *is given by*

$$f_k(1) = \begin{cases} 0 & \text{if } k = 0,\\ 3 & \text{if } k = 1,\\ 2^k + 2^{k-1} + 1 & \text{if } 2 \le k \le n-1. \end{cases}$$

Given a graph $G = (V, E)$, a *matching M* of G is a set of pairwise non-adjacent edges of G. A *perfect matching* is a matching that saturates all the vertices; in other words, every vertex in the graph is incident to exactly one edge in the matching. From Eq. (1), for all vertices x of LTQ_n and for all $0 \le k \le n-1$, we have

$$f_k(f_k(x)) = x. \tag{2}$$

Therefore, for a fixed k, the set of edges connecting a vertex and its k-th dimensional neighbor forms a perfect matching of LTQ_n. More precisely,

$$M_k = \{(x, f_k(x) \mid x \in V(LTQ_n))\}$$

is a perfect matching of LTQ_n. See Fig. 2 for an illustration.

2.3. The even–odd-vertex-transitivity of the locally twisted cube

A graph is vertex-transitive if for every pair of vertices u and v, there is an automorphism that maps u to v. Intuitively, a vertex-transitive network looks the same from every node. The vertex-transitive property is advantageous to the design and simulation of some algorithms. It is not difficult to see that LTQ_2 and LTQ_3 are vertex-transitive; see Fig. 1. However, in the following, we will show that LTQ_n is not vertex-transitive when $n \ge 4$.

Theorem 2.2. *The locally twisted cube* LTQ_n *is not vertex-transitive for* $n \ge 4$.

Proof. For $n = 4$, let $N_k(r)$ denote the set $N_k(r) = \{x \in V(LTQ_n) \mid d(x, r) = k\}$. Consider the set $\Omega(r) = \{x \in N_2(r) \mid N_1(x) \cap N_1(r) = 1 \text{ and } N_1(x) \cap N_3(r) = 1\}$. Then $\Omega(0) = \{7\}$, but $\Omega(1) = \{6, 12\}$; see Fig. 3 for an illustration. Therefore LTQ_4 is not vertex-transitive.

Now consider LTQ_n with $n \ge 5$. It is well-known that vertices 0 and $2^n - 2$ are at the farthest distance of LTQ_n and $d(2^n - 2, 0) = \lceil \frac{n+3}{2} \rceil$. In the following, we prove that LTQ_n is not vertex-transitive by showing the following claim.

Claim 2.3. *For an arbitrary vertex $x \in V(LTQ_n)$, $n \geq 5$, the distance $d(x, 1) \leq \lceil \frac{n+1}{2} \rceil$.*

Proof of Claim 2.3. Before showing the claim, some notations are introduced first. Let $x = x_{n-1}x_{n-2}\dots x_0$. Scanning the bits of x from x_{n-1} to x_1 (notice that we ignore the bit x_0). Suppose there are a total of m bits equal to 1 and a total of k disjoint pairs of consecutive bits equal to "11", we denoted it by "11"-bits. A bit x_i, $1 \leq i \leq n-1$, is said to be *isolated* if after removing the k disjoint pairs of "11"-bits of x, we have $x_i = 1$. For example, consider $x = 111011$ in LTQ_6. Then $m = 4$, $k = 1$ and x_1, x_3 are isolated. Clearly, $0 \leq k \leq \lfloor \frac{m}{2} \rfloor$ holds.

It should be noticed that if $m < \lceil \frac{n-1}{2} \rceil$, then there exists a trivial path from x to 1: (i) If $x_0 = 0$, then corrects all $x_i = 1$ bits, $1 \leq i \leq n-1$, to 0, and then corrects x_0 to 1; (ii) If $x_0 = 1$, then corrects x_0 to 0. Then corrects all $x_i = 1$ bits, $1 \leq i \leq n-1$, to 0, and then correct x_0 to 1. Clearly, both paths have length at most $m + 2 \leq \lceil \frac{n+1}{2} \rceil$. In the following, we assume $m \geq \lceil \frac{n-1}{2} \rceil$. Therefore,

$$m - \left\lceil \frac{n-1}{2} \right\rceil \leq k \leq \left\lfloor \frac{m}{2} \right\rfloor$$

holds. There are two cases.

Case 1: $x_0 = 0$. A path from x to 1 can be found as follows: Step 1: Remove all the isolated bits of x. Step 2: Correct x_0 to 1. Step 3: Match all "11"-bits. Clearly, Steps 1, 2 and 3 take $m - 2k$, 1 and k steps, respectively. The total number of steps is

$$m - k + 1 \leq m - \left(m - \left\lceil \frac{n-1}{2} \right\rceil\right) + 1 = \left\lceil \frac{n+1}{2} \right\rceil.$$

For example, consider $x = 11101010$ in LTQ_8. We have $m = 5$, $k = 1$ and x_1, x_3, x_5 are isolated bits. A path from x to 1 is built as follows: $11101010 \xrightarrow{\text{Step 1}} 11001010 \xrightarrow{\text{Step 1}} 11000010 \xrightarrow{\text{Step 1}} 11000000 \xrightarrow{\text{Step 2}} 11000001 \xrightarrow{\text{Step 3}} 00000001$.

Case 2: $x_0 = 1$. We further divide this case into two subcases:

Subcase 2.1: $m + 1 - \lceil \frac{n-1}{2} \rceil \leq k \leq \lfloor \frac{m}{2} \rfloor$. Then a path from x to 1 can be found as follows: Step 1: Correct x_0 to 0. Step 2: Remove all the isolated bits of x. Step 3: Correct x_0 to 1. Step 4: Match all "11"-bits. Clearly, Steps 1, 2, 3 and 4 take 1, $m - 2k$, 1 and k steps, respectively. Thus the total number of steps is

$$m - k + 2 \leq \left\lceil \frac{n+1}{2} \right\rceil.$$

For example, consider $x = 11011011$ in LTQ_8. We have $m = 5$, $k = 2$ and x_1 is a isolated bit. A path from x to 1 is built as follows: $11011011 \xrightarrow{\text{Step 1}} 11011010 \xrightarrow{\text{Step 2}} 11011000 \xrightarrow{\text{Step 3}} 11011001 \xrightarrow{\text{Step 4}} 00011001 \xrightarrow{\text{Step 4}} 00000001$.

Subcase 2.2: $k = m - \lceil \frac{n-1}{2} \rceil$. In this case, all bits $x_{n-1}, x_{n-3}, \dots, x_1$ must equal to 1 if n is even; either all bits $x_{n-2}, x_{n-3}, \dots, x_1$ or all bits $x_{n-1}, x_{n-3}, \dots, x_2$ must equal to 1 if n is odd. Thus a path from x to 1 can be found by bitwise correcting the bits to 0 (by scanning the bits from x_{n-1} to x_1). Since it takes one step to correct an isolated bit and one step to correct a "11"-bits, the total step is

$$(m - 2k) + k = \left\lceil \frac{n-1}{2} \right\rceil.$$

For example, consider $x = 10111011$ in LTQ_8. We have $m = 5$, $k = 1$. A path from x to 1 is built as follows: $10111011 \xrightarrow{\text{isolated}} 01111011 \xrightarrow{\text{“11”-bits}} 00011011 \xrightarrow{\text{“11”-bits}} 00000011 \xrightarrow{\text{isolated}} 00000001$. □

From the above discussion, we have $d(x, 1) \leq \lceil \frac{n+1}{2} \rceil$. As a result, LTQ_n is not vertex-transitive for $n \geq 4$. □

Although LTQ_n fails to be vertex-transitive for $n \geq 4$, it does satisfy the *even–odd-vertex-transitive* property: for every pair of vertices $x = x_{n-1}x_{n-2}\dots x_0$, $y = y_{n-1}y_{n-2}\dots y_0$ with the same parity, i.e., $x_0 = y_0$, there is an automorphism ψ that maps x to y. In other words, in LTQ_n, all even-numbered vertices are symmetric and all odd-numbered vertices are symmetric. By using this property, we may pay our attention of constructing ISTs to use vertex 0 and vertex 1 as the common root without loss of generality.

Theorem 2.4. *The locally twisted cube LTQ_n satisfies the even–odd-vertex-transitive property.*

Proof. It suffices to prove that there exists an automorphism which maps v ($\neq 0$) to 0 (resp., v ($\neq 1$) to 1), whenever v is an even-numbered (resp., odd-numbered) vertex. For two n-bits binary strings x and y, let $x \oplus y$ denote the bitwise XOR (modulo 2) of x and y. Let $v = v_{n-1}v_{n-2}\dots v_0 \in V(LTQ_n)$.

Suppose v is an even-numbered vertex. For $x = x_{n-1}x_{n-2}\dots x_0 \in V(LTQ_n)$, define a function ψ_0 as follows:

$$\psi_0(x) = v \oplus x.$$

It is not difficult to see that ψ_0 is a bijection from $V(LTQ_n)$ to $V(LTQ_n)$. Now we verify that ψ_0 preserves the adjacency. Consider any edge $(x, f_k(x)) \in E(LTQ_n)$. Since $v_0 = 0$, we have

$$\psi_0(x) = (v_{n-1} \oplus x_{n-1})\,(v_{n-2} \oplus x_{n-2})\,\dots(v_{k+1} \oplus x_{k+1})\,(v_k \oplus x_k)\,(v_{k-1} \oplus x_{k-1})\,\dots\,(v_1 \oplus x_1)\,x_0.$$

Algorithm 1 CONSTRUCT_IST

Input: All vertices of LTQ_n and root r.
Output: n ISTs $T_0, T_1, \ldots, T_{n-1}$ rooted at r.
1: **for** $i = 0$ to $n - 1$ **do in parallel** ▷ construct T_i simultaneously
2: $\quad$ $child\_of\_the\_root \leftarrow f_i(r)$
3: $\quad$ $V(T_i) \leftarrow \{child\_of\_the\_root\}$
4: $\quad$ **for** $t = 1$ to n **do** ▷ outer for-loop
5: $\quad\quad$ $S \leftarrow \emptyset$;
6: $\quad\quad$ **for** each vertex $v \in V(T_i)$ **do** ▷ inner for-loop
7: $\quad\quad\quad$ $u \leftarrow f_{(i+t) \bmod n}(v)$
8: $\quad\quad\quad$ $E(T_i) \leftarrow E(T_i) \cup \{(v, u)\}$ ▷ set the parent of vertex u as v in T_i
9: $\quad\quad\quad$ $S \leftarrow S \cup \{u\}$
10: $\quad\quad$ **end for**
11: $\quad\quad$ $V(T_i) \leftarrow V(T_i) \cup S$
12: $\quad$ **end for**
13: **end for**

Also,

$$\psi_0(f_k(x)) = \begin{cases} (v_{n-1} \oplus x_{n-1})\,(v_{n-2} \oplus x_{n-2})\,\ldots\,(v_1 \oplus u_1)\,\bar{x}_0 & \text{if } k = 0,\\ (v_{n-1} \oplus x_{n-1})\,(v_{n-2} \oplus x_{n-2})\,\ldots\,(v_2 \oplus u_2)\,(v_1 \oplus \bar{x}_1)\,x_0 & \text{if } k = 1, \end{cases}$$

and for $2 \le k \le n - 1$,

$$\psi_0(f_k(x)) = (v_{n-1} \oplus x_{n-1})\,(v_{n-2} \oplus x_{n-2})\,\ldots\,(v_{k+1} \oplus x_{k+1})\,(v_k \oplus \bar{x}_k)\,(v_{k-1} \oplus x_{k-1} \oplus x_0)\,(v_{k-2} \oplus x_{k-2})\,\ldots\,(v_1 \oplus x_1)\,x_0.$$

Since $v_k \oplus \bar{x}_k = \overline{v_k \oplus x_k}$ no matter $v_k = x_k$ or $v_k \neq x_k$, we have

$$\psi_0(f_k(x)) = f_k(\psi_0(x))$$

and hence $(\psi_0(x), \psi_0(f_k(x))) \in E(LTQ_n)$.

Similar arguments can be applied to the case of v being an odd-numbered vertex, except that the bijection function from $V(LTQ_n)$ to $V(LTQ_n)$ is replaced by

$$\psi_1(x) = v \oplus x \oplus 1. \quad \square$$

3. The algorithm

We now present an algorithm, called CONSTRUCT_IST, for constructing n ISTs $T_0, T_1, \ldots, T_{n-1}$ rooted at an arbitrary vertex r for the locally twisted cube LTQ_n in Algorithm 1. For convenience, call the for-loop in lines 4–12 of this algorithm the "outer for-loop" and call the for-loop in lines 6–10 the "inner for-loop". This algorithm constructs $T_0, T_1, \ldots, T_{n-1}$ simultaneously and it works as follows. Since LTQ_n is n-regular, the n neighbors of the root r must be the unique child of the root r in $T_0, T_1, \ldots, T_{n-1}$, respectively. In this algorithm, the unique child of the root r in T_i is set as $f_i(r)$. Thus, initially $V(T_i) = \{f_i(r)\}$. At the tth iteration of the outer for-loop, each vertex v in $V(T_i)$ is connected to a new vertex $u = f_{(i+t) \bmod n}(v)$ by using the edges in perfect matching $M_{(i+t) \bmod n}$, and the edge (v, u) is added to T_i (i.e., the parent of u is set as v in T_i). After n iterations of the outer for-loop, T_i is constructed.

Example 1. We now demonstrate how Algorithm CONSTRUCT_IST constructs T_2 rooted at vertex 1 in LTQ_4. In line 2 of the algorithm, the unique child of the root 1 is set as $f_2(1) = 7$. Thus $V(T_2) = \{7\}$. Now consider the outer for-loop. For $t = 1$, each vertex in $V(T_2)$ is connected to a new vertex by using the edges in M_3; thus the edge (7, 11) is added to T_2; so S becomes {11} and $V(T_2)$ becomes {7, 11}. For $t = 2$, each vertex in $V(T_2)$ is connected to a new vertex by using the edges in M_0; thus the edges (7, 6) and (11, 10) are added to T_2; so S becomes {6, 10} and $V(T_2)$ becomes {7, 11, 6, 10}. For $t = 3$, each vertex in $V(T_2)$ is connected to a new vertex by using the edges in M_1; thus the edges (7, 5), (11, 9), (6, 4) and (10, 8) are added to T_2; so S becomes {5, 9, 4, 8} and $V(T_2)$ becomes {7, 11, 6, 10, 5, 9, 4, 8}. Finally, for $t = 4$, each vertex in $V(T_2)$ is connected to a new vertex by using the edges in M_2; thus the edges (7, 1), (11, 13), (6, 2), (10, 14), (5, 3), (9, 15), (4, 0) and (8, 12) are added to T_2; so S becomes {1, 13, 2, 14, 3, 15, 0, 12} and $V(T_2)$ becomes {7, 11, 6, 10, 5, 9, 4, 8, 1, 13, 2, 14, 3, 15, 0, 12}. See Fig. 4 for an illustration.

4. Correctness

The purpose of this section is to prove that $T_0, T_1, \ldots, T_{n-1}$ generated by Algorithm Construct_IST are n ISTs rooted at an arbitrary vertex r for LTQ_n. To this end, some notations are first introduced in Section 4.1. We show that $T_0, T_1, \ldots, T_{n-1}$ are n spanning trees of LTQ_n in Section 4.2. The vertex-independency of $T_0, T_1, \ldots, T_{n-1}$ is shown in Section 4.3.

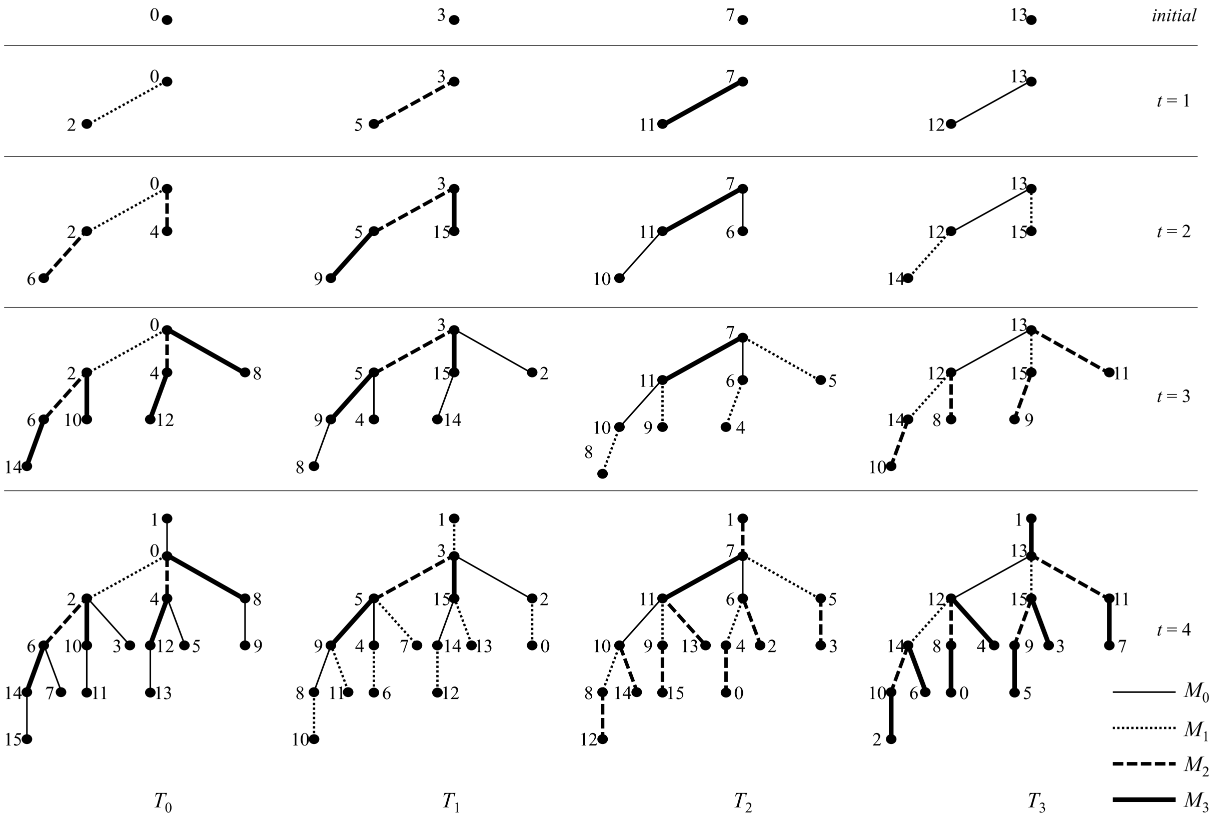

Fig. 4. Four ISTs rooted at vertex 1 in LTQ_4 constructed by Algorithm Construct_IST.

4.1. The notations

Definition 3. For $V' \subseteq V(LTQ_n)$, define $f_i(V')$ to be

$$f_i(V') = \{f_i(v) \mid v \in V'\}.$$

Definition 4. For a fixed integer i, $0 \le i \le n-1$, define O_i^n to be the *ordered* set

$$O_i^n = \{i, (i-1) \bmod n, (i-2) \bmod n, \ldots, (i-n+1) \bmod n\}.$$

Notice that O_i^n can be obtained by arranging $0, 1, \ldots, n-1$ around a circle, starting from the number i and picking up these n numbers counterclockwise. For example, $O_0^4 = \{0, 3, 2, 1\}$, $O_1^4 = \{1, 0, 3, 2\}$ and $O_3^4 = \{3, 2, 1, 0\}$.

Definition 5. The *Hamming distance* between two vertices $x, y \in V(LTQ_n)$, denoted by $Ham(x, y)$, is the number of positions at which the corresponding symbols are different. More precisely, $Ham(x, y) = |\{i \mid x_i \ne y_i, 0 \le i \le n-1\}|$. For two fixed vertices $x, y \in V(LTQ_n)$, suppose $Ham(x, y) = m$. Define $H_i(x, y)$ to be an *ordered* set consisting of the indices of the m different bits, listed according to the order given by O_i^n.

Definition 6. For two fixed vertices $x, y \in V(LTQ_n)$, suppose $H_i(x, y) = \{c_{m-1}, c_{m-2}, \ldots, c_0\}$ with $m \ge 2$ and $H_i(x, y) \ne O_i^n$. We say that *j is between c_u and c_{u-1} for some $0 \le u \le m-1$ with respect to O_i^n* if $j \notin H_i(x, y)$ and when $0, 1, \ldots, n-1$ are arranged on a circle, the location of j on the circle is between c_u and c_{u-1}.

For example, consider LTQ_4. Suppose $v = 12$. Then $H_0(v, 0) = \{3, 2\}$, $H_1(v, 3) = \{1, 0, 3, 2\}$, $H_2(v, 7) = \{1, 0, 3\}$ and $H_3(v, 13) = \{0\}$. Since $1 \notin H_0(v, 0)$, 1 is between $c_u = 3$, $c_{u-1} = 2$; $0 \notin H_0(v, 0)$, 0 is between $c_u = 2$, $c_{u-1} = 3$.

Definition 7. For two vertices $x, y \in V(LTQ_n)$, define $\Pi_i(x, y)$ to be the *ordered* set consisting of all the indices of perfect matchings used in the x, y-path in T_i, $0 \le i \le n-1$, listed according to the order from x to y.

For example, consider T_2 rooted at vertex 1 of LTQ_4 in Fig. 4. Suppose $v = 12$. Then $\Pi_2(v, 7) = \{2, 1, 0, 3\}$. Moreover, the path from v to 7 is

$$1100 \xrightarrow{M_2} 1000 \xrightarrow{M_1} 1010 \xrightarrow{M_0} 1011 \xrightarrow{M_3} 0111.$$

Definition 8. Define $I(a, b)$, where $a \ge b$, to be the sequence such that

$$I(a, b) = \begin{cases} a, a-1, \ldots, b+1 & \text{if } a > b, \\ a & \text{if } a = b. \end{cases}$$

4.2. The spanning trees

Throughout this subsection, let $T_0, T_1, \ldots, T_{n-1}$ be the output of Algorithm Construct_IST. The purpose of this subsection is to prove that $T_0, T_1, \ldots, T_{n-1}$ are n spanning trees rooted at r. By Theorem 2.4, we assume $r = 0$ and $r = 1$ as the common roots without loss of generality. To prove that T_i, $0 \le i \le n-1$, is a spanning tree rooted at r, we prove the following *loop invariant*:

Loop invariant: At the start of the tth iteration of the outer for-loop, T_i is connected, $|V(T_i)| = 2^{t-1}$ and $|E(T_i)| = |V(T_i)| - 1$.

The loop invariant is trivial true prior to the first loop iteration since in line 3, Algorithm Construct_IST sets $V(T_i) = \{f_i(r)\}$. Hence T_i is connected, $|V(T_i)| = 2^0$ and $|E(T_i)| = |V(T_i)| - 1$. We now prove that if the loop invariant is true before the tth iteration of the outer for-loop, then it remains true before the next iteration. Algorithm Construct_IST first resets S to be empty in line 5. For each vertex v in $V(T_i)$, Algorithm Construct_IST adds the edge (v, u) to T_i in line 8, where $u = f_{(i+t) \bmod n}(v)$, by using the edges in $M_{(i+t) \bmod n}$, and adds u to S in line 9. Since each newly generated edge is incident to a vertex in $V(T_i)$, T_i remains to be connected. Now we claim that

Claim 4.1. *$V(T_i) \cap S = \emptyset$.*

If Claim 4.1 is true, then at the end of the inner for-loop, the newly generated edges between $V(T_i)$ and S clearly form a matching that saturates $V(T_i)$ and S. Thus $|V(T_i)| = |S|$. Consequently, after the tth iteration of the outer for-loop, T_i is connected, $|V(T_i)| = 2^{t-1} + 2^{t-1} = 2^t$ and $|E(T_i)| = 2^{t-1} - 1 + 2^{t-1} = 2^t - 1 = |V(T_i)| - 1$. When the outer for-loop terminates, $t = n + 1$. Therefore, T_i is connected, $|V(T_i)| = 2^n$ and $|E(T_i)| = |V(T_i)| - 1$. Also, at the end of the $(t = n)$th iteration of the outer for-loop, Algorithm Construct_IST adds the edge $(r, f_i(r))$ to T_i. Therefore T_i is a spanning tree rooted at r of LTQ_n. In the following, we prove that Claim 4.1 is true for $r = 0$ and $r = 1$. We first consider the case of $r = 0$.

Lemma 4.2. *Claim 4.1 is true for $r = 0$.*

Proof. Consider the tth iteration of the outer for-loop. Set $k = (i + t) \bmod n$ for easy writing. Let $v \in V(T_i)$ and $u \in S$. If $t \in \{1, 2, \ldots, n-1\}$, then $(v_k, u_k) = (0, 1)$. If $t = n$, then we have $(v_i, u_i) = (1, 0)$. Therefore $V(T_i) \cap S = \emptyset$. □

Lemma 4.3. *Claim 4.1 is true for $r = 1$.*

Proof. Consider T_i, $0 \le i \le n-1$. Set $k = (i + t) \bmod n$ for easy writing. Let $v \in V(T_i)$ and $u \in S$.

Case 1: $i = 0$. If $t \in \{1, 2, \ldots, n-1\}$, then $(v_k, u_k) = (0, 1)$. If $t = n$, then $(v_i, u_i) = (1, 0)$. Therefore $V(T_i) \cap S = \emptyset$.

Case 2: $i = n - 1$. If $t \in \{1, 2, \ldots, n-2\}$, then $(v_k, u_k) = (0, 1)$. If $t = n - 1$, then we have $(v_{n-2}, u_{n-2}) = (1, 0)$. If $t = n$, then we have $(v_i, u_i) = (1, 0)$. Therefore $V(T_i) \cap S = \emptyset$.

Case 3: $i \in \{1, 2, \ldots, n-2\}$. We further divide this case into two subcases.

Subcase 3.1: $t \in \{1, 2, \ldots, n-2\}$. The proof of this case is the same as Case 2.

Subcase 3.2: $t = n$. By the loop invariant, T_i induces a tree before the tth iteration of the outer for-loop. Partition $V(T_i)$ into V_0 and V_1 as follows:

$$V_0 = \{\text{all the vertices in the subtree rooted at } f_{i+1}(f_i(1))\} \quad \text{and} \quad V_1 = V(T_i) \setminus V_0.$$

See Fig. 5 for an illustration.

By (1) and by Lemma 4.6, we have: (i) the ith bit of all the vertices in V_0 is 0 and hence the ith bit of all the vertices in $f_i(V_0)$ is 1, and (ii) the ith bit of all the vertices in V_1 is 1 and hence the ith bit of all the vertices in $f_i(V_1)$ is 0. Notice that

$$S = f_i(V_0) \cup f_i(V_1).$$

Therefore, to prove Claim 4.1, it suffices to prove that

$$V_0 \cap f_i(V_1) = \emptyset \quad \text{and} \quad V_1 \cap f_i(V_0) = \emptyset. \tag{3}$$

If $i = n - 2$, then the $(n-1)$-bit of all the vertices in V_0 and $f_{n-2}(V_0)$ is 1; however, the $(n-1)$-bit of all the vertices in V_1 and $f_{n-2}(V_1)$ is 0. Thus when $i = n-2$, $V_0 \cap f_{n-2}(V_1) = \emptyset$ and $V_1 \cap f_{n-2}(V_0) = \emptyset$. Now suppose $i \in \{1, 2, \ldots, n-3\}$. Partition V_0 into $V_{0,0}$ and $V_{0,1}$ such that

$$V_{0,0} = \{\text{all the vertices in the subtree rooted at } f_{i+2}(f_{i+1}(f_i(1)))\} \quad \text{and} \quad V_{0,1} = V_0 \setminus V_{0,0}.$$

Partition V_1 into $V_{1,0}$ and $V_{1,1}$ such that

$$V_{1,0} = \{\text{all the vertices in the subtree rooted at } f_{i+2}(f_i(1))\} \quad \text{and} \quad V_{1,1} = V_1 \setminus V_{1,0}.$$

By (1) and Lemma 4.6, the pair of the $(i+1)$th and the ith bit of all the vertices in $V_{0,0}$ and $f_i(V_{1,1})$ is $(0, 0)$; in $f_i(V_{0,0})$ and $V_{1,1}$ is $(0, 1)$; in $V_{0,1}$ and $f_i(V_{1,0})$ is $(1, 0)$ and in $f_i(V_{0,1})$ and $V_{1,0}$ is $(1, 1)$. Thus to prove (3), it suffices to prove that

$$V_{0,0} \cap f_i(V_{1,1}) = \emptyset, \qquad V_{1,1} \cap f_i(V_{0,0}) = \emptyset, \qquad V_{1,0} \cap f_i(V_{0,1}) = \emptyset \quad \text{and} \quad V_{0,1} \cap f_i(V_{1,0}) = \emptyset. \tag{4}$$

For $v = v_{n-1}, v_{n-1}, \ldots, v_0 \in V(LTQ_n)$ with $v \neq 0$, let q be the largest index of v such that $v_q = 1$. If $v = 0$, then let $q = -1$. By (1) and Lemma 4.6, we have Table 2.

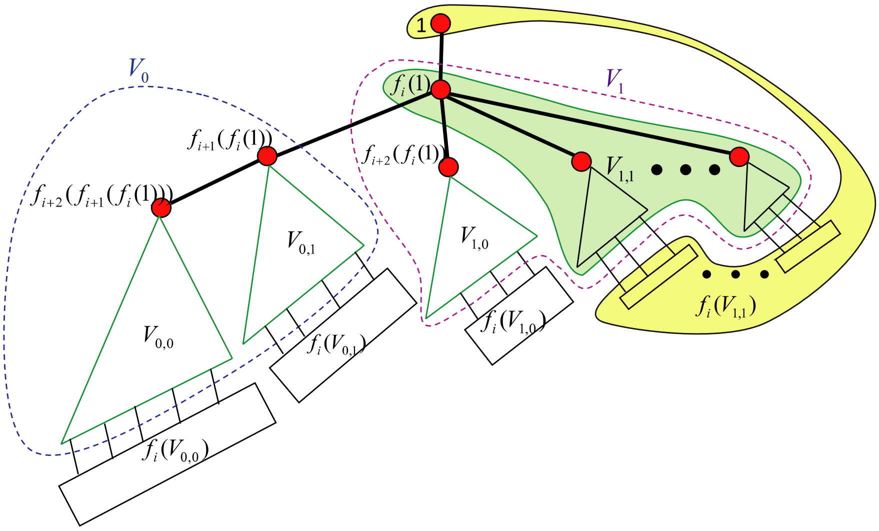

Fig. 5. An illustration for the proof of Lemma 4.3.

Table 2
The value of q for every vertex in the given set.

$V_{0,0} \cup f_i(V_{0,0})$	$V_{1,1} \cup f_i(V_{1,1})$	$V_{1,0} \cup f_i(V_{1,0})$	$V_{0,1} \cup f_i(V_{0,1})$
$q \geq i+2$	$q \leq i+1$ or $q \geq i+3$	$q \geq i+3$	$q = i+1$ or $q \geq i+3$

We first prove that $V_{0,0} \cap f_i(V_{1,1}) = \emptyset$ and $V_{1,1} \cap f_i(V_{0,0}) = \emptyset$. By Table 2, each vertex in $V_{1,1} \cap f_i(V_{1,1})$ with $q \leq i+1$ does not belong to $V_{0,0} \cup f_i(V_{0,0})$ since every vertex in $V_{0,0} \cup f_i(V_{0,0})$ has $q \geq i+2$. Also, each vertex in $V_{0,0} \cup f_i(V_{0,0})$ with $q = i+2$ does not belong to $V_{1,1} \cap f_i(V_{1,1})$ since each vertex in $V_{1,1} \cap f_i(V_{1,1})$ has $q \neq i+2$. Thus, we may focus on vertices with $q = i+3$ or $q > i+3$. Note that each vertex in $V_{0,0} \cup f_i(V_{0,0})$ with $q = i+3$ has its $(i+2)$th bit to be 0; however, from Table 2, we know that each vertex in $f_i(V_{1,1}) \cup V_{1,1}$ with $q \geq i+3$ has its $(i+2)$th bit to be 1. Therefore, each vertex in $V_{0,0} \cup f_i(V_{0,0})$ with $q = i+3$ does not belong to $V_{1,1} \cup f_i(V_{1,1})$. It remains to consider the vertices with $q > i+3$. For each $x \in V_{0,0} \cup f_i(V_{0,0})$, the bit string of x formed by x_q to x_{i+2} is in

$$L_0 = \{\underbrace{1\overbrace{00\cdots0}^{q-i-2\text{ 0's}}}_{q-i-1\text{ bits}}, \underbrace{1\overbrace{00\cdots0}^{q-i-4\text{ 0's}}11}_{q-i-1\text{ bits}}, \underbrace{1\overbrace{00\cdots0}^{q-i-5\text{ 0's}}101}_{q-i-1\text{ bits}}, \underbrace{1\overbrace{00\cdots0}^{q-i-6\text{ 0's}}1001}_{q-i-1\text{ bits}}, \ldots, \underbrace{101\overbrace{00\cdots0}^{q-i-5\text{ 0's}}1}_{q-i-1\text{ bits}}, \underbrace{11\overbrace{00\cdots0}^{q-i-4\text{ 0's}}1}_{q-i-1\text{ bits}}\}.$$

However, for each $y \in V_{1,1} \cup f_i(V_{1,1})$, the bit string of y formed by y_q to y_{i+2} is in

$$L_1 = \{\underbrace{1\overbrace{00\cdots0}^{q-i-3\text{ 0's}}1}_{q-i-1\text{ bits}}, \underbrace{1\overbrace{00\cdots0}^{q-i-4\text{ 0's}}10}_{q-i-1\text{ bits}}, \underbrace{1\overbrace{00\cdots0}^{q-i-5\text{ 0's}}100}_{q-i-1\text{ bits}}, \underbrace{1\overbrace{00\cdots0}^{q-i-6\text{ 0's}}1000}_{q-i-1\text{ bits}}, \ldots, \underbrace{101\overbrace{00\cdots0}^{q-i-4\text{ 0's}}}_{q-i-1\text{ bits}}, \underbrace{11\overbrace{00\cdots0}^{q-i-3\text{ 0's}}}_{q-i-1\text{ bits}}\}.$$

It is not difficult to check that $L_0 \cap L_1 = \emptyset$. Hence we have $V_{0,0} \cap f_i(V_{1,1}) = \emptyset$ and $V_{1,1} \cap f_i(V_{0,0}) = \emptyset$.

Similar arguments can show that $V_{0,1} \cap f_i(V_{1,0}) = \emptyset$ and $V_{1,0} \cap f_i(V_{0,1}) = \emptyset$, except that $V_{0,0} \cup f_i(V_{0,0})$ is replaced by $V_{1,0} \cup f_i(V_{1,0})$ and $V_{1,1} \cup f_i(V_{1,1})$ is replaced by $V_{0,1} \cup f_i(V_{0,1})$. From the above discussion, we have (4) and hence have (3). Therefore $V(T_i) \cap S = \emptyset$. □

By Theorem 2.4 and Lemmas 4.2 and 4.3, we have the following result.

Lemma 4.4. *$T_0, T_1, \ldots, T_{n-1}$ are n spanning trees rooted at r for LTQ_n.*

4.3. The vertex-independency of the n spanning trees

In this subsection, we show that $T_0, T_1, \ldots, T_{n-1}$ generated by Algorithm Construct_IST are vertex-independent trees rooted at an arbitrary vertex r for LTQ_n. By Theorem 2.4, without loss of generality, we may assume $r = 0$ and $r = 1$ as the common roots. To this end, we need to show that for any i, j with $0 \leq i < j \leq n-1$ and for each $v(\neq r) \in V(LTQ_n)$, the r, v-path in T_i and the r, v-path in T_j are internally vertex-disjoint. Recall that the child of the root in T_i and T_j are $f_i(r)$ and $f_j(r)$, respectively. In the following, we further assume $v \notin \{r, f_i(r), f_j(r)\}$ since if $v \in \{r, f_i(r), f_j(r)\}$, then the r, v-path in T_i and the r, v-path in T_j are clearly internally vertex-disjoint. Let $parent_i(v)$ (resp., $parent_j(v)$) be the parent of vertex v in T_i

(resp., T_j). Let P_1 (resp., P_2) be the $parent_i(v), f_i(r)$-path (resp., $parent_j(v), f_j(r)$-path) in T_i (resp., T_j). Since $f_i(r) \neq f_j(r)$, the r, v-path in T_i and the r, v-path in T_j are internally vertex-disjoint if and only if $V(P_1) \cap V(P_2) = \emptyset$. We prove T_i and T_j are vertex-independent by showing the following claim:

Claim 4.5. $V(P_1) \cap V(P_2) = \emptyset$.

Before proving Claim 4.5, we need a lemma.

Lemma 4.6. T_i, $0 \leq i \leq n-1$, *constructed by Algorithm Construct_IST has the property that for each* $v \in V(LTQ_n) \setminus \{r, f_i(r)\}$, *the path from* v *to* $f_i(r)$ *in* T_i *uses each perfect matching in* $\{M_0, M_1, \ldots, M_{n-1}\}$ *at most once.*

Proof. It follows from the fact that $f_{(i+t) \bmod n}$ used in the for-loop between the inner for-loop are distinct when the outer for-loop iterates from $t = 1$ to $t = n$. □

We first consider the case of $r = 0$.

Lemma 4.7. $T_0, T_1, \ldots, T_{n-1}$ *are* n *vertex-independent trees rooted at* $r = 0$ *for* LTQ_n.

Proof. To prove Claim 4.5, we first describe the path from v to the child of the root in T_i when $r = 0$. For any $v \in V(T_i) \setminus \{0, f_i(0)\}$, the $v, f_i(0)$-path in T_i can be determined by $\Pi_i(v, f_i(0))$. In addition, $\Pi_i(v, f_i(0))$ can be determined by $H_i(v, f_i(0))$ as follows. Suppose $v = v_{n-1}v_{n-2}\ldots v_0$ and $H_i(v, f_i(0)) = \{c_{m-1}, c_{m-2}, \ldots, c_0\}$. If $v_0 = 0$, then $\Pi_i(v, f_i(0))$ can be determined by

$$\Pi_i(v, f_i(0)) = \begin{cases} H_i(v, f_i(0)) & \text{if } i \neq 0, \\ \{c_{m-1} = 0, I(c_{m-2}, c_{m-3}), \ldots, I(c_3, c_2), I(c_1, c_0)\} & \text{if } i = 0 \text{ and } m-1 \text{ is even}, \\ \{c_{m-1} = 0, I(c_{m-2}, c_{m-3}), \ldots, I(c_2, c_1), I(c_0, 0)\} & \text{if } i = 0 \text{ and } m-1 \text{ is odd}. \end{cases} \tag{5}$$

If $v_0 = 1$ and $i \neq 0$, then $H_i(v, f_i(0))$ must contain 0; in this case, we assume $c_e = 0$ for some e. Thus if $v_0 = 1$, $\Pi_i(v, f_i(0))$ can be determined by

$$\Pi_i(v, f_i(0)) = \begin{cases} \{I(c_{m-1}, c_{m-2}), I(c_{m-3}, c_{m-4}), \ldots, I(c_1, c_0)\} & \text{if } i = 0 \text{ and } m \text{ is even}, \\ \{I(c_{m-1}, c_{m-2}), I(c_{m-3}, c_{m-4}), \ldots, I(c_2, c_1), I(c_0, 0)\} & \text{if } i = 0 \text{ and } m \text{ is odd}, \\ \{I(c_{m-1}, c_{m-2}), I(c_{m-3}, c_{m-4}), \ldots, I(c_{e+2}, c_{e+1}), c_e, c_{e-1}, \ldots, c_0\} & \text{if } i \neq 0 \text{ and } m-e \text{ is odd}, \\ \{I(c_{m-1}, c_{m-2}), I(c_{m-3}, c_{m-4}), \ldots, I(c_{e+1}, 0), c_e, c_{e-1}, \ldots, c_0\} & \text{if } i \neq 0 \text{ and } m-e \text{ is even}. \end{cases} \tag{6}$$

Now we show that Claim 4.5 is true for $r = 0$. Suppose not, then there exists a vertex a $(\neq v) \in V(P_1) \cap V(P_2)$. Suppose

$$H_i(v, f_i(0)) = H_i(v, 2^i) = \{c_{m-1}, c_{m-2}, \ldots, c_0\}. \tag{7}$$

There are four cases.

Case 1: $v_i = 1$ and $v_j = 1$. Then there must exist u such that $c_u = j$. Thus

$$H_j(v, f_j(0)) = H_j(v, 2^j) = \{c_{u-1}, c_{u-2}, \ldots, c_0, i, c_{m-1}, c_{m-2}, \ldots, c_{u+1}\}. \tag{8}$$

By (5)–(7), c_{m-1} is the first element in $\Pi_i(v, 2^i)$. Let $x \in V(P_1)$. Then the (c_{m-1})th bit of x is $v_{c_{m-1}}$ only when (i) $(c_{m-1}+1) \in \Pi_i(v, 2^i)$, and (ii) $c_{m-1}+1 \geq 2$, and (iii) there exists $q = q_{n-1}q_{n-2}\ldots q_0 \in V(P_1)$ such that $x = f_{c_{m-1}+1}(q)$ and $q_0 = 1$. We now prove that (i)–(iii) will not occur simultaneously; hence for all $x \in V(P_1)$, the (c_{m-1})th bit of x is $\overline{v}_{c_{m-1}}$. If $|H_i(v, 2^i)| = 1$, then (i) cannot occur. Suppose $|H_i(v, 2^i)| \geq 2$ and both (i) and (iii) occur; that is, there exists $q = q_{n-1}q_{n-2}\ldots q_0 \in V(P_1)$ such that $x = f_{c_{m-1}+1}(q)$ and $q_0 = 1$. By (7), $c_{m-1}+1$ is the last element in $\Pi_i(v, 2^i)$. Since $q_0 = 1$, $I(c_0, 0) \subseteq \Pi_i(v, 2^i)$. By Lemma 4.6 and by the fact that $I(c_0, 0) = \{c_0, c_0 - 1, \ldots, 1\}$, we have $c_{m-1}+1 = 1$; thus (ii) does not occur and consequently the (c_{m-1})th bit of all the vertices in $V(P_1)$ is $\overline{v}_{c_{m-1}}$. Since $v_i = 1$, the ith bit of all the vertices in $V(P_1)$ is 1. By (5) and (6) and (8), the (c_{m-1})th bit of those vertices in $V(P_2)$ with the ith bit being 1 is $v_{c_{m-1}}$. Thus no such a exists and Claim 4.5 is true.

Case 2: $v_i = 0$ and $v_j = 0$. Then $c_{m-1} = i$. If $|H_i(v, 2^i)| = 1$, then $H_i(v, 2^i) = \{i\}$, which implies that $v = 0$; this contradicts to the assumption that $v \neq 0$. Thus $|H_i(v, 2^i)| \geq 2$ and there must exist u such that j is between c_u and c_{u-1} with respect to O_i^n. Thus

$$H_j(v, 2^j) = \begin{cases} \{j, c_{m-2}, c_{m-3}, \ldots, c_{u+1}, c_{u=0}\} & \text{if } j = i+1, \\ \{j, c_{u-1}, c_{u-2}, \ldots, c_0, c_{m-2}, c_{m-3}, \ldots, c_{u+1}\} & \text{if otherwise}. \end{cases} \tag{9}$$

By (5)–(7), the ith bit of all vertices in $V(P_1)$ is 1. By (5) and (6) and (9), the jth bit of all the vertices in $V(P_2)$ is 1. The ith bit and the jth bit of a are both 1. If $I(c_u, c_{u-1}) \not\subseteq \Pi_i(v, 2^i)$, each vertex in $V(P_1)$ has its jth bit to be 0. If (i) $j \neq i+1$ and $I(c_0, c_{m-2}) \not\subseteq \Pi_j(v, 2^j)$, or if (ii) $j = i+1$ and $v_0 \neq 1$, then each vertex in $V(P_2)$ has its ith bit to be 0. Thus the existence of a implies that $I(c_u, c_{u-1}) \subseteq \Pi_i(v, 2^i)$ and $I(c_0, c_{m-2}) \subseteq \Pi_j(v, 2^j)$. Note that $I(c_u, c_{u-1}) \subseteq \Pi_i(v, 2^i)$ implies that $i = 0$ and hence $v_0 = 0$ (since case 2 requires $v_i = 0$). However, $I(c_0, c_{m-2}) \subseteq \Pi_j(v, 2^j)$ implies $v_0 = 1$, which contradicts to $v_0 = 0$. Thus no such a exists and Claim 4.5 is true.

Case 3: $v_i = 0$ and $v_j = 1$. Then $c_{m-1} = i$ and there must exist u such that $c_u = j$. If $|H_i(v, 2^i)| = 1$, then $H_j(v, 2^i) = \emptyset$. This implies that $v = 2^j$, which contradicts to the assumption that $v \neq 2^j$. Thus

$$H_j(v, 2^j) = \{c_{u-1}, c_{u-2}, \ldots, c_0, c_{m-2}, c_{m-3}, \ldots, c_{u+1}\}. \tag{10}$$

By (5)–(7), the ith bit of all vertices in $V(P_1)$ is 1. The ith bit of a is 1. If $I(c_0, c_{m-2}) \nsubseteq \Pi_j(v, 2^j)$, each vertex in $V(P_2)$ has its ith bit to be 0. Thus the existence of a implies that $I(c_0, c_{m-2}) \subseteq \Pi_j(v, 2^j)$, which further implies $v_0 = 1$. Since $I(c_0, c_{m-2}) \subseteq \Pi_j(v, 2^j)$, $V(P_2)$ has only one vertex $x = x_{n-1}x_{n-2}\ldots x_0$ such that $x_i = 1$ and $x = f_{i+1}(q)$ for some $q \in V(P_2)$. The existence of a implies that $x = a$. Since $v_0 = 1$, $\Pi_i(v, 2^i)$ starts with $I(i, c_{m-2})$, i.e., $\Pi_i(v, 2^i)$ is of the form $\{I(i, c_{m-2}), \ldots\}$. By (6), c_{m-3} is the first element after $I(i, c_{m-2})$ in $\Pi_i(v, 2^i)$. Recall that $\Pi_i(v, 2^i)$ is an ordered set of all the indices of perfecting matchings used in the $v, 2^i$-path in T_i listed according to the order from v to 2^i. Thus the first vertex in $V(P_1)$ can be obtained by applying the first perfect matching obtained from the first element in $\Pi_i(v, 2^i)$ to v, the second vertex in $V(P_1)$ can be obtained by applying the second perfect matching obtained from the second element in $\Pi_i(v, 2^i)$ to the first vertex in $V(P_1)$, and so on. Thus we can partition $V(P_1)$ into $V_{1,1}$ and $V_{1,2}$ such that $V_{1,1}$ consists of those vertices in $V(P_1)$ before $f_{c_{m-3}}$ is applied and $V_{1,2} = V(P_1) - V_{1,1}$. Let $y = y_{n-1}y_{n-2}\ldots y_0$ be an arbitrary vertex in $V_{1,1}$. Then $Ham(y_iy_{i-1}\ldots y_{c_{m-2}}, v_iv_{i-1}\ldots v_{c_{m-2}}) = 2$. However, $Ham(x_ix_{i-1}\ldots x_{c_{m-2}}, v_iv_{i-1}\ldots v_{c_{m-2}}) = 0$. Thus $x \notin V_{1,1}$. On the other hand, $x_{c_{m-3}} = v_{c_{m-3}}$ but the (c_{m-3})th bit of all the vertices in $V_{1,2}$ is $\overline{v}_{c_{m-3}}$; thus $x \notin V_{1,2}$. Since $x \notin V_{1,1}$ and $x \notin V_{1,2}$, we have $x \notin V(P_1)$. Since $x = a$, it follows that $a \notin V(P_1)$. Thus no such a exists and Claim 4.5 is true.

Case 4: $v_i = 1$ and $v_j = 0$. Then there must exist u such that j is between c_u and c_{u-1} with respect to O_i^n. Thus

$$H_j(v,2^j) = \begin{cases} \{j, i, c_{m-1}, c_{m-2}, \ldots, c_{u=0}\} & \text{if } i \text{ is between } c_0 \text{ and } c_{m-1} \text{ with respect to } O_i^n, \\ \{j, c_{u-1}, c_{u-2}, \ldots, c_0, i, c_{m-1}, c_{m-2}, \ldots, c_u\} & \text{if otherwise.} \end{cases} \tag{11}$$

By (5), (6) and (11), the jth bit of all vertices in $V(P_2)$ is 1. Since $v_i = 1$, the ith bit of all the vertices in $V(P_1)$ is 1. The ith bit and the jth bit of a are both 1. By (11), we have two subcases.

Subcase 4.1: i is between c_0 and c_{m-1} with respect to O_i^n. Then $V(P_2)$ has only one vertex $f_j(v)$ with its ith bit and jth bit both being 1. By (5)–(7), c_{m-1} is the first element in $\Pi_i(v, 2^i)$. Thus the (c_{m-1})th bit of those vertices in $V(P_1)$ with the jth bit being 1 is $\overline{v}_{c_{m-1}}$. However, by (5), (6) and (11), the (c_{m-1})th bit of $f_j(v)$ is $v_{c_{m-1}}$. Thus no such a exists and Claim 4.5 is true.

Subcase 4.2: i is not between c_0 and c_{m-1} with respect to O_i^n. By (5), (6) and (11), the ith bit of all the vertices in $V(P_1)$ is 1. If $|H_i(v, 2^i)| = 1$, then $H_i(v, 2^i) = \{c_0\}$; since $v_j = 0$, we have $c_0 \neq j$, which implies that each vertex in $V(P_1)$ has its jth bit to be 0 and consequently no such a exists and Claim 4.5 is true. Now suppose $|H_i(v, 2^i)| \geq 2$. Then when $I(c_u, c_{u-1}) \nsubseteq \Pi_i(v, 2^i)$, each vertex in $V(P_1)$ has its jth bit to be 0. Thus the existence of a implies that $I(c_u, c_{u-1}) \subseteq \Pi_i(v, 2^i)$. Since $I(c_u, c_{u-1}) \subseteq \Pi_i(v, 2^i)$, $V(P_1)$ has only one vertex $x = x_{n-1}x_{n-2}\ldots x_0$ such that $x_j = 1$ and $x = f_{j+1}(q)$ for some $q \in V(P_1)$. The existence of a implies that $x = a$. By (5), (6) and (11), the (c_{m-1})th bit of those vertices in $V(P_2)$ with the ith bit being 1 is $v_{c_{m-1}}$. However, the $x_{c_{m-1}} = \overline{v}_{c_{m-1}}$. So if $x \in V(P_1)$, then $x \notin V(P_2)$. Thus no such a exists and Claim 4.5 is true. □

From the above discussion, Claim 4.5 is true and therefore $T_0, T_1, \ldots, T_{n-1}$ are vertex-independent rooted at $r = 0$ of LTQ_n. □

Now we consider the case of $r = 1$.

Lemma 4.8. *$T_0, T_1, \ldots, T_{n-1}$ are n vertex-independent trees rooted at $r = 1$ for LTQ_n.*

Proof. To prove Claim 4.5, we first describe the path from v to the child of the root in T_i when $r = 1$. For any $v \in V(T_i) \setminus \{1, f_i(1)\}$, the $v, f_i(1)$-path in T_i can be determined by $\Pi_i(v, f_i(1))$. Furthermore, $\Pi_i(v, f_i(1))$ can be determined by the ordered set $H_i(v, f_i(1))$ as follows. Suppose $v = v_{n-1}v_{n-2}\ldots v_0$ and $H_i(v, f_i(1)) = \{c_{m-1}, c_{m-2}, \ldots, c_0\}$. Let c_{e-1} be the first (from bit c_{m-1} to c_0) member in $H_i(v, f_i(1))$ that is larger than i. If $i = 0$, $\Pi_i(v, f_i(1))$ can be determined by

$$\Pi_0(v, f_0(1)) = H_0(v, f_0(1)). \tag{12}$$

If $i \neq 0$ and $v_0 = 0$, we have $c_e = 0$ for some e. Thus $\Pi_i(v, f_i(1))$ can be determined by

$$\Pi_i(v, f_i(1)) = \begin{cases} \{c_{m-1}, c_{m-2}, \ldots, c_e, I(c_{e-1},c_{e-2}), I(c_{e-3},c_{e-4}), \ldots, I(c_1, c_0)\} & \text{if } e \text{ is even,} \\ \{c_{m-2}, c_{m-3}, \ldots, c_e, I(c_{e-1},c_{e-2}), I(c_{e-3},c_{e-4}), \ldots, I(c_0, i)\} & \text{if } e \text{ is odd and } c_{m-1} = i, \\ \{i, c_{m-1}, c_{m-2}, \ldots, c_e, I(c_{e-1},c_{e-2}), I(c_{e-3},c_{e-4}), \ldots, I(c_0, i)\} & \text{if } e \text{ is odd and } c_{m-1} \neq i. \end{cases} \tag{13}$$

When $i \neq 0$ and $v_0 = 1$, in order to obtain $\Pi_i(v, f_i(1))$ from $H_i(v, f_i(1))$, the following notations are introduced. Define H_i^1 to be the sequence

$$H_i^1 = \begin{cases} c_{m-1}, c_{m-2}, \ldots, c_e & \text{if } |H_i^2| \text{ is even,} \\ i, c_{m-1}, c_{m-2}, \ldots, c_e & \text{if } |H_i^2| \text{ is odd and } c_{m-1} \neq i \\ c_{m-2}, c_{m-3}, \ldots, c_e & \text{if } |H_i^2| \text{ is odd and } c_{m-1} = i, \end{cases}$$

and define H_i^2 to be the sequence

$$H_i^2 = c_{e-1}, c_{e-2}, \ldots, c_0.$$

Define $\zeta_i(v, f_i(1))$ to be the sequence

$$\zeta_i(v, f_i(1)) = \begin{cases} H_i^1, H_i^2 & \text{if } |H_i^1| \text{ is even and } |H_i^2| \text{ is even,} \\ H_i^1, H_i^2, i & \text{if } |H_i^1| \text{ is even and } |H_i^2| \text{ is odd,} \\ H_i^1, 0, H_i^2 & \text{if } |H_i^1| \text{ is odd and } |H_i^2| \text{ is even,} \\ H_i^1, 0, H_i^2, i & \text{if } |H_i^1| \text{ is odd and } |H_i^2| \text{ is odd.} \end{cases} \tag{14}$$

Suppose

$$\zeta_i(v, f_i(1)) = \varsigma_u, \varsigma_{u-1}, \ldots, \varsigma_0.$$

Thus if $i \neq 0$ and $v_0 = 1$, $\Pi_i(v, f_i(1))$ can be determined by

$$\Pi_i(v, f_i(1)) = \{I(\varsigma_u, \varsigma_{u-1}), I(\varsigma_{u-2}, \varsigma_{u-3}), \ldots, I(\varsigma_1, \varsigma_0), \}. \tag{15}$$

Now we show that Claim 4.5 is true for $r = 1$. Suppose not, then there exists a vertex $a\ (\neq v) \in V(P_1) \cap V(P_2)$. Suppose

$$H_i(v, f_i(1)) = \{c_{m-1}, c_{m-2}, \ldots, c_0\}. \tag{16}$$

There are four cases.

Case 1: $0 = i < j \leq n - 1$. The proof of this case is divided into two parts, depending on $v_0 = 1$ or $v_0 = 0$. Suppose $v_0 = 1$. Then $0 \notin H_j(v, f_j(1))$. Thus the 0th bit of all the vertices in $V(P_2)$ is 1. By (12) and (16), 0 is the first element in $H_0(v, f_0(1))$; this implies that the 0th bit of all the vertices in $V(P_1)$ is 0. Thus no such a exists. In the following, we assume $v_0 = 0$. Then $0 \notin H_0(v, f_0(1))$. The 0th bit of all the vertices in $V(P_1)$ is 0; this implies that the 0th bit of a is 0. There are two possibilities: $j = 1$ or $j > 1$.

Subcase 1.1: $j = 1$. Note that either $1 \in \Pi_1(v, f_1(1))$ or $1 \notin \Pi_1(v, f_1(1))$. If $1 \notin \Pi_1(v, f_1(1))$, then 0 is the first element in $\Pi_1(v, f_1(1))$. This implies that the 0th bit of all the vertices in $V(P_2)$ is 1. Thus no such a exists. If $1 \in \Pi_1(v, f_1(1))$, then 1 and 0 are the first element and the second element in $\Pi_1(v, f_1(1))$, respectively. Thus the 0th bit of all the vertices in $V(P_2) \setminus \{f_1(v)\}$ is 1. The existence of a implies that $f_1(v) = a$.

If $v_1 = 0$, then $1 \notin H_0(v, f_0(1))$. This implies that the 1st bit of all the vertices in $V(P_1)$ is 0. However, it is obvious that the 1st bit of $f_1(v)$ is 1. Therefore $f_1(v) \notin V(P_1)$. Thus no such a exists. Now suppose $v_1 = 1$. Since $1 \in \Pi_1(v, f_1(1))$, there must exist some $k > 1$ such that $v_k = 1$; this implies that $c_{m-1} > 1$. By (12) and (16), the (c_{m-1})th bit of all the vertices in $V(P_1)$ is $\overline{v}_{c_{m-1}}$. However, the (c_{m-1})th bit of $f_1(v)$ is $v_{c_{m-1}}$. Therefore $f_1(v) \notin V(P_1)$. Thus no such a exists and $V(P_1) \cap V(P_2) = \emptyset$.

Subcase 1.2: $j > 1$. By (12), (13) and (16), we have: c_{m-1} is the first element in $H_i(v, f_i(1))$, $c_{m-1} \in H_j(v, f_j(1))$, $0 \in H_j(v, f_j(1))$, and c_{m-1} appears after 0 in the ordered set $H_j(v, f_j(1))$. Thus the (c_{m-1})th bit of all the vertices in $V(P_1)$ is $\overline{v}_{c_{m-1}}$. However, the (c_{m-1})th bit of those vertices with the 0th bit being 0 in $V(P_2)$ is $v_{c_{m-1}}$. Thus no such a exists.

From the above discussion, Claim 4.5 is true for *Case* 1.

Case 2: $1 = i < j \leq n - 1$. The proof of this case is divided into two parts, depending on $v_0 = 0$ or $v_0 = 1$.

Subcase 2.1: $v_0 = 0$. Then it is not difficult to see (by comparing the jth and the 0th bits of $f_j(v)$ and all the vertices in $V(P_1)$) that $f_j(v) \notin V(P_1)$. Thus a can not be $f_j(v)$. It remains to consider those vertices in $V(P_2) \setminus f_j(v)$. The remaining proof is further divided into two parts, depending on $v_{j-1} = 0$ or $v_{j-1} = 1$.

Subcase 2.1.1: $v_{j-1} = 0$. Since $v_0 = 0$ and $v_{j-1} = 0$, $j - 1 \in \Pi_j(v, f_j(v))$. Since $v_0 = 0$ and $j - 1 \in \Pi_j(v, f_j(v))$, the $(j - 1)$th bit of all the vertices in $V(P_2) \setminus f_j(v)$ is 1. However, the $(j - 1)$th bit of all the vertices in $V(P_1)$ is 0. Thus no such a exists and Claim 4.5 is true.

Subcase 2.1.2: $v_{j-1} = 1$. We claim that: the bits from v_{j-2} to v_2 are all 0, i.e., $v_{j-2} = v_{j-3} = \cdots = v_2 = 0$. Suppose this claim is not true and let k be the largest number between $j - 2$ and 2 (inclusive) such that $v_k = 1$. By (13) and (16), the $(j - 1)$th and the kth bits of all the vertices in $V(P_2) \setminus f_j(v)$ is 1 and 0, respectively. However, the $(j - 1)$th bit of those vertices in $V(P_1)$ with kth bit being 0 is 0. Thus $v_{j-2} = v_{j-3} = \cdots = v_2 = 0$. So the 1st bit of all the vertices in $V(P_1)$ is 1 and the 1st bit of all the vertices in $V(P_2) \setminus f_j(v)$ is 0. Thus no such a exists and Claim 4.5 is true.

Subcase 2.2: $v_0 = 1$. The proof of this part is further divided into six parts as follows.

Subcase 2.2.1: $j = 2$, $v_1 = 1$ and $v_2 = 1$. Since $v_0 = 1$ and $v_1 = 1$ and $v_2 = 1$,

$$H_j(v, f_j(1)) = (c_{m-1}, c_{m-2}, \ldots, c_1).$$

Suppose m is even. Then by (14) and (15),

$$\Pi_i(v, f_i(1)) = \{I(c_{m-1}, c_{m-2}), \ldots, I(c_1, c_0 = 2)\}$$

and

$$\Pi_j(v, f_j(1)) = \{I(2, 0), I(c_{m-1}, c_{m-2}), \ldots, I(c_1, 2)\}.$$

Thus, the 2nd bit of all the vertices in $V(P_1)$ are 1. However, the 2nd bit of all the vertices in $V(P_2)$ are 0. Thus no such a exists. Suppose m is odd. Then by (14) and (15),

$$\Pi_i(v, f_i(1)) = \{1, I(c_{m-1}, c_{m-2}), \ldots, I(c_0, 1)\}$$

and

$$\Pi_j(v, f_j(1)) = \{I(c_{m-1}, c_{m-2}), \ldots, I(c_2, c_1)\}.$$

Hence the 1st bit of all the vertices in $V(P_1)$ is 0. However, the 1st bit of all the vertices in $V(P_2)$ is 1. Thus no such a exists.

Subcase 2.2.2: $j = 2$, $v_1 = 0$ and $v_2 = 1$. Since $v_0 = 1$ and $v_1 = 0$ and $v_2 = 1$, we have $c_{m-1} = 1$, $c_0 = 2$ and

$$H_j(v, f_j(1)) = \{c_{m-1}, c_{m-2}, \ldots, c_1\}.$$

Suppose $m - 1$ is odd. Then by (14) and (15),

$$\Pi_i(v, f_i(1)) = \{I(c_{m-2}, c_{m-3}), \ldots, I(c_0, 1)\}$$

and

$$\Pi_j(v, f_j(1)) = \{1, I(c_{m-2}, c_{m-3}), \ldots, I(c_2, c_1)\}.$$

Thus, the 1st bit of all vertices in $V(P_1)$ are 0. However, the 1st bit of all vertices in $V(P_2)$ is 1. Thus no such a exists. Suppose $m - 1$ is even. Then by (14) and (15),

$$\Pi_i(v, f_i(1)) = \{1, I(c_{m-2}, c_{m-3}), \ldots, I(c_1, c_0)\}$$

and

$$\Pi_j(v, f_j(1)) = \{2, I(c_{m-2}, c_{m-3}), \ldots, I(c_1, 2)\}.$$

Thus, the 2nd bit of all vertices in $V(P_1)$ are 1. However, the 2nd bit of all vertices in $V(P_2)$ is 0. Thus no such a exists.

Subcase 2.2.3: $j = 2$, $v_1 = 1$ and $v_2 = 0$ (resp., $v_1 = 0$ and $v_2 = 0$). Then

$$H_j(v, f_j(1)) = \{2, c_{m-1}, c_{m-2}, \ldots, c_0\}.$$

Suppose m (resp., $m - 1$) is even. Then by (14) and (15), the 2nd bit of all vertices in $V(P_1)$ is 0. However, the 2nd bit of all vertices in $V(P_2)$ is 1. Suppose m (resp., $m - 1$) is odd. Then by (14) and (15), the 1st bit of all vertices in $V(P_1)$ is 0. However, the 1st bit of all vertices in $V(P_2)$ is 1. Thus no such a exists.

Subcase 2.2.4: $j \neq 2$ and $v_{j-1} = 0$. Then the $(j-1)$th bit of all the vertices in $V(P_1)$ are 0. However, the $(j-1)$th bit of all the vertices in $V(P_2)$ are 1. Thus no such a exists.

Subcase 2.2.5: $j \neq 2$, $v_{j-1} = 1$ and at least one of the bits in $v_{j-2}v_{j-3}\ldots v_2$ is 1. Then there exists q such that $v_q = 1$ and q is the largest number between $j - 2$ and 2 (inclusive).

Subcase 2.2.5.1: Suppose $I(j, q) \nsubseteq \Pi_j(v, f_j(1))$. Then the qth and the $(j-1)$th bit of all the vertices in $V(P_2)$ are 0 and 1, respectively; however, the $(j-1)$th bit of those vertices in $V(P_1)$ with the qth bit being 0 is 0. Thus no such a exists.

Subcase 2.2.5.2: Suppose $I(j, q) \subseteq \Pi_j(v, f_j(1))$. Then we partition $V(P_2)$ into $V_{2,1}$ and $V_{2,2}$ such that

$$V_{2,1} = \{\text{all the vertices in } V(P_2) \text{ before the perfect matching } M_q \text{ is applied}\} \text{ and } V_{2,2} = V(P_2) \setminus V_{2,1}.$$

Consider the vertices in $V_{2,1}$. Suppose $v_j = 0$. Since $j \in I(j, q)$, we can compare the jth bit of all vertices in $V(P_1)$ and in $V_{2,1}$ to see that no such a exists. Suppose $v_j = 1$. Then the number of bits in $v_{n-1}v_{n-2}\ldots v_{j+1}$ that are 1 is odd, i.e., $|H_j^2|$ is odd. This implies that $c_{m-1} \neq j$. Since $c_{m-1} \neq j$, by comparing the c_{m-1}th bit of all the vertices in $V(P_1)$ and in $V_{2,1}$, we know that $V(P_1) \cap V_{2,1} = \emptyset$. Consider the vertices in $V_{2,2}$. Then the qth and the $(j-1)$th bit of all the vertices in $V_{2,2}$ are 0 and 1, respectively. However, the $(j-1)$th bit of those vertices in $V(P_1)$ with the qth bit being 0 is 0. Hence $V(P_1) \cap V_{2,2} = \emptyset$. Since $V(P_1) \cap V_{2,1} = \emptyset$ and $V(P_1) \cap V_{2,2} = \emptyset$, no such a exists.

Subcase 2.2.6: $j \neq 2$, $v_{j-1} = 1$ and all the bits in $v_{j-2}v_{j-3}\dots v_2$ are 0 (i.e., $v_{j-2} = v_{j-3} = \cdots = v_2 = 0$). For convenience, let $t(w_1, w_2)$ denote the number of bits in $v_{w_1}v_{w_1-1}\dots v_{w_2}$ that are 1. There are three possibilities.

Subcase 2.2.6.1: Suppose $t(n-1, i+1)$ is even. Then $t(n-1, j)$ is odd. Thus the ith bit of all the vertices in $V(P_2)$ is 0. However, the ith bit of all the vertices in $V(P_1)$ is 1. Thus no such a exists.

Subcase 2.2.6.2: Suppose $t(n-1, i+1)$ is odd and $v_j = 0$. Then $t(n-1, j+1)$ is even. Thus the jth bit of all the vertices in $V(P_2)$ is 1. However, the jth bit of all the vertices in $V(P_1)$ is 0. Thus no such a exists.

Subcase 2.2.6.3: Suppose $t(n-1, i+1)$ is odd and $v_j = 1$. Then $t(n-1, j+1)$ is odd. Thus the ith bit of all the vertices in $V(P_1)$ is 0 and the jth bit of all the vertices in $V(P_2)$ is 0. Then the. ith and the jth bit of a are 0. By (15), the $(j-1)$th bit of all the vertices in $V(P_2)$ with the ith and the jth bit be 0 is 1. However, only the vertex $2^{j-1}+1$ in $V(P_1)$ with the $(j-1)$th bit is 1, and the ith and the jth bit are 0. The existence of a implies $a = 2^{j-1}+1$. Since $t(n-1, j+1)$ is odd, there exists $v_k = 1$, where $k > j$. Then it is easy to find that $a \notin V(P_2)$ by comparing the $k$thBthe jth and the ith bit of a and all vertices in $V(P_2)$. Thus no such a exists.

From the above discussion, Claim 4.5 is true for *Case* 2.

Case 3: $3 \leq i+1 = j \leq n-1$. By (12)–(16), we have the following results. Suppose $t(n-1, i+1)$ is odd. Then the ith bit of all vertices in $V(P_1)$ is 0 and $j \notin \Pi_j(v, f_j(1))$; however, the ith bit of all the vertices in $V(P_2)$ is 1. Suppose $t(n-1, i+1)$ is even and $v_j = 0$. Then the jth bit of all the vertices in $V(P_2)$ is 1; however, the jth bit of all the vertices in $V(P_1)$ is 0. Suppose $t(n-1, i+1)$ is even and $v_j = 1$. Then the jth bit of all the vertices in $V(P_2)$ is 0; however, the jth bit of all the vertices in $V(P_1)$ is 1. Thus no such a exists and Claim 4.5 is true.

Case 4: $3 \leq i+1 < j \leq n-1$. We divide the proof into three parts, depending on the values of v_{j-1} and v_{i-1}.

Subcase 4.1: $v_{j-1} = 0$. Then if $j \in \Pi_i(v, f_i(1))$, then $V(P_1)$ has only one vertex (say, vertex x) with its $(j-1)$th bit being 1. By comparing from the jth to the $(i-1)$th bits of x with the jth to the $(i-1)$th bits of each vertex in $V(P_2)$, we have $x \notin V(P_2)$. If $j \in \Pi_j(v, f_j(1))$, then $f_j(v)$ is the unique vertex in $V(P_2)$ with its $(j-1)$th bit being 0. By comparing from the jth to the $(i-1)$th bits of $f_j(v)$ with the jth to the $(i-1)$th bits of each vertex in $V(P_1)$, we have $f_j(v) \notin V(P_1)$. Then by (12)–(16), the $(j-1)$th bit of all the vertices in $V(P_1) \setminus \{x\}$ is 0; however, the $(j-1)$th bit of all the vertices in $V(P_2) \setminus f_j(v)$ is 1. Thus no such a exists.

Subcase 4.2: $v_{i-1} = 0$. Then we can use similar arguments to prove that no such a exists.

Subcase 4.3: $v_{i-1} = 1$ and $v_{j-1} = 1$. By (12)–(15), we have following the results. When $i \in H_i(v, f_i(1))$ and $v_0 = 1$, $V(P_1)$ has only one vertex $f_i(v)$ with the $(i-1)$th bit being 0. It is easy to find $f_i(v) \notin V(P_2)$ by comparing those bits from the $(j-1)$th to the $(i-1)$th of $f_i(v)$ with each vertex in $V(P_2)$. And since the $(i-1)$th bit of all the vertices in $V(P_1) \setminus f_i(v)$ is 1, the existence of a implies that the $(i-1)$th bit of a must be 1.

Partition $V(P_2)$ into two $V_{2,1}$ and $V_{2,2}$ such that

$$V_{2,1} = \{\text{all the vertices in } V(P_2) \text{ before the perfect matching } M_i \text{ is applied}\} \text{ and } V_{2,2} = V(P_2) \setminus V_{2,1}.$$

Thus the $(i-1)$th bit of all the vertices in $V_{2,1}$ is 1, and if a exists, then $a \in V_{2,1}$. We now claim that:

Claim 4.9. *If a exists, then* $v_{j-2} = v_{j-3} = \cdots = v_{i+1} = 0$.

Proof of Claim 4.9. Suppose this claim is not true. Then let q be the largest index between $j-2$ and $i+1$ (inclusive) such that $v_q = 1$. Let $y = y_{n-1}y_{n-2}\dots y_0$ be an arbitrary vertex in $V_{2,1} \setminus \{f_j(v)\}$. Note that $f_j(v) \in V_{2,1}$ only when $j \in \Pi_j(v, f_j(1))$. Also note that $q \in \Pi_j(v, f_j(1))$. Moreover, if $j \in H_j(v, f_j(1))$, then q is the first element after j in $H_j(v, f_j(1))$; if $j \notin H_j(v, f_j(1))$, then q is the first element in $H_j(v, f_j(1))$. Since q exists, by (13)–(15), the bits $y_{j-2}y_{j-3}\dots y_{i+1}$ will be different from the bits $v_{j-2}v_{j-3}\dots v_{i+1}$. However, let $x = x_{n-1}x_{n-2}\dots x_0$ be an arbitrary vertex in $V(P_1)$. Then the bits $x_{j-2}x_{j-3}\dots x_{i+1}$ are identical to the bits $v_{j-2}v_{j-3}\dots v_{i+1}$. Thus every vertex in $V_{2,1} \setminus \{f_j(v)\}$ is not in $V(P_1)$. Although $f_j(v) \in V_{2,1}$, $f_j(v)$ is not in $V(P_1)$ (this can be observed by comparing the jth bit and from the $(j-2)$th to the $(i+1)$th bits of all the vertices in $V(P_1)$ with jth bit and the bits from the $(j-2)$th to the $(i+1)$th bits of $f_j(v)$). Thus $V(P_1) \cap V_{2,1} = \emptyset$. Since if a exists, then $a \in V_{2,1}$. Thus a does not exists and we have this claim. □

By Claim 4.9, in the remaining proof, we assume $v_{i-1} = 1$, $v_{j-1} = 1$ and $v_{j-2} = v_{j-3} = \cdots = v_{i+1} = 0$. For convenience, let t denote the number of bits in $v_{n-1}v_{n-2}\dots v_{j+1}$ that are 1. We further divided the proof into four subcases.

Subcase 4.3.1: $v_i = 1$ and $v_j = 1$. Suppose t is even. Then the first member in $\Pi_j(v, f_j(1))$ is i. However, $i \notin \Pi_i(v, f_i(1))$. Thus no such a exists and $V(P_1) \cap V(P_2) = \emptyset$. Suppose t is odd. Then $j \in \Pi_j(v, f_j(1))$ and $I(j-1, i) \subset \Pi_i(v, f_i(1))$. Thus the jth bit of all the vertices in $V(P_2)$ is 0. Partition $V(P_1)$ into $V_{1,1}$ and $V_{1,2}$ such that

$$V_{1,1} = \{\text{all the vertices in } V(P_1) \text{ before the perfect matching } M_{(j+1) \bmod n} \text{ is applied}\} \text{ and } V_{1,2} = V(P_1) \setminus V_{1,1}.$$

Thus the jth bit of all vertices in $V_{1,1}$ is 1 and the jth bit of all vertices in $V_{1,2}$ is 0. By the fact that the jth bit of all the vertices in $V(P_2)$ is 0, to prove $V(P_1) \cap V(P_2) = \emptyset$, it suffices to prove $V_{1,2} \cap V(P_2) = \emptyset$. If $v_0 = 1$, then the $(j-1)$th bit of all the vertices in $V(P_2) \setminus f_j(v)$ is 1; however, the $(j-1)$th bit of all the vertices in $V_{1,2}$ is 0. Since the ith bit of is 1, but the ith bit of

all the vertices in $V_{1,2}$ is 0, $f_j(v) \notin V_{1,2}$. If $v_0 = 0$, then the $(j-1)$th bit of all the vertices in $V(P_2)$ is 1, and the $(j-1)$th bit of all the vertices in $V_{1,2} \setminus \{z = 2^{j-1} + 2^{i-1} + 1\}$ is 0. Since t is odd, there exists $v_k = 1$ for some $k > j$. Thus $z \notin V(P_2)$ by comparing the kth bit of them. Therefore, no such a exists in this case.
Subcase 4.3.2: $v_i = 0$ and $v_j = 0$. Suppose t is even. Then the jth bit of all the vertices in $V(P_2)$ is 1. However, the jth bit of all the vertices in $V(P_1)$ is 0. Suppose t is odd. Then the number of bits in $v_{n-1}v_{n-2}\ldots v_{i+1}$ that are 1 is even; this implies that i is the first member in $\Pi_i(v, f_i(1))$. Thus the ith bit of all the vertices in $V(P_2)$ is 0. However, the ith bit of all the vertices in $V(P_1)$ is 1. Thus no such a exists.

Subcase 4.3.3: $v_i = 0$ and $v_j = 1$. Suppose t is even. Then the first member in $\Pi_j(v, f_j(1))$ is $i-1$ and the first member in $\Pi_i(v, f_i(1))$ is i. So the ith bit of all the vertices in $V(P_2)$ is 0; however, the ith bit of all the vertices in $V(P_1)$ is 1. Suppose t is odd. Define q to be the index of the leftmost nonzero bit of v. Then $q > j$. Thus the $(i-1)$th bit of all the vertices in $V(P_2) \setminus \{f_j(v)\}$ is 0; however, the $(i-1)$th bit of all the vertices in $V(P_1)$ is 1. By comparing the jth and the qth bits of $f_j(v)$ with the jth and the qth bits of every vertex in $V(P_1)$, we have $f_j(v) \notin V(P_1)$. Thus no such a exists.

Subcase 4.3.4: $v_i = 1$ and $v_j = 0$. If the number of those bits from v_{n-1} to v_{j+1} being 1 is even, then the jth bit of all the vertices in $V(P_2)$ is 1; however the jth bit of all the vertices in $V(P_1)$ is 0. If the number of those bits from v_{n-1} to v_{j+1} being 1 is odd, then the number of bits in $v_{n-1}v_{n-2}\ldots v_{i+1}$ that are 1 is even. Thus i is the first member of $\Pi_j(v, f_j(1))$; but $i \notin \Pi_i(v, f_j(1))$, which implies that the ith bit of all the vertices in $V(P_2)$ is 0 but the ith bit of all the vertices in $V(P_1)$ is 1. So Claim 4.5 is true for this case.

As a result, Claim 4.5 is true for *Case* 4. From the above discussion, Claim 4.5 is true for all the cases, and therefore $T_0, T_1, \ldots, T_{n-1}$ are vertex-independent rooted at $r = 0$ of LTQ_n. $\square$

By Theorem 2.4 and Lemmas 4.7 and 4.8, we have the following result.

Theorem 4.10. *$T_0, T_1, \ldots, T_{n-1}$ are n vertex-ISTs rooted at r for LTQ_n.*

5. Concluding remarks

The independent spanning trees (ISTs) problem attempts to construct a set of pairwise independent spanning trees and it has numerous applications in networks such as data broadcasting, scattering and reliable communication protocols. The well-known ISTs conjecture, Vertex/Edge Conjecture, states that any n-connected/n-edge-connected graph has n vertex-ISTs/edge-ISTs rooted at an arbitrary vertex r. Both the Vertex and Edge Conjectures are still open on general graphs for $n \geq 5$.

In this paper, we consider the ISTs problem on the n-dimensional locally twisted cube LTQ_n. The very recent algorithm proposed by Hsieh and Tu [12] is designed to construct n edge-ISTs rooted at vertex 0 for LTQ_n. However, we find that LTQ_n is not vertex-transitive when $n \geq 4$ and therefore Hsieh and Tu's result does not solve the Edge Conjecture for LTQ_n. In this paper, we present an algorithm to construct n vertex-independent spanning trees rooted at an arbitrary vertex for LTQ_n. To the best of our knowledge, this is the first result to confirm the Vertex Conjecture for the locally twisted cubes. In addition, it is also interesting to confirm whether the Vertex Conjecture is true for other hypercube variants.

References

[1] F. Bao, Y. Funyu, Y. Hamada, Y. Igarashi, Reliable broadcasting and secure distributing in channel networks, IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences E81A (1998) 796–806.
[2] F. Bao, Y. Igarashi, S.R. Ohring, Reliable broadcasting in product networks, Graph-theoretic Concepts in Computer Science 1517 (1998) 310–323.
[3] Y.S. Chen, C.Y. Chiang, C.Y. Chen, Multi-node broadcasting in all-ported 3-D wormhole-routed torus using an aggregation-then-distribution strategy, Journal of System Architecture 50 (2004) 575–589.
[4] G. Chartrand, L. Lensniak, Graph and Digraphs, Wadsworth, Monterey, CA, 1981.
[5] S. Curran, O. Lee, X. Yu, Finding four independent trees, SIAM Journal on Computing 35 (2006) 1023–1058.
[6] J. Cheriyan, S.N. Maheshwari, Finding nonseparating induced cycles and independent spanning trees in 3-connected graphs, Journal of Algorithms 9 (1988) 507–537.
[7] J. Edmonds, Edge-disjoint branchings, in: R. Rustin (Ed.), Combinatorial Algorithms, in: Courant Inst. Sci. Symp., vol. 9, Algorithmics Press, New York, 1973, pp. 91–96.
[8] Z. Ge, S.L. Hakimi, Disjoint rooted spanning trees with small depths in de Bruijn and Kautz graphs, SIAM Journal on Computing 26 (1997) 79–92.
[9] A. Huck, Independent trees in planar graphs, Graphs and Combinatorics 5 (1999) 29–77.
[10] A. Huck, Independent trees in graphs, Graphs and Combinatorics 10 (1994) 29–45.
[11] T. Hasunuma, H. Nagamochi, Independent spanning trees with small depths in iterated line digraphs, Discrete Applied Mathematics 110 (2001) 189–211.
[12] S.Y. Hsieh, C.J. Tu, Constructing edge-disjoint spanning trees in locally twisted cubes, Theoretical Computer Science 410 (2009) 8–10.
[13] K.S. Hu, S.S. Yeoh, C.Y. Chen, L.H. Hsu, Node-pancyclicity and edge-pancyclicity of hypercube variants, Information Processing Letters 102 (1) (2007) 1–7.
[14] Y. Iwasaki, Y. Kajiwara, K. Obokata, Y. Igarashi, Independent spanning trees of chordal rings, Information Processing Letters 69 (1999) 155–160.
[15] A. Itai, M. Rodeh, The multi-tree approach to reliability in distributed networks, Information and Computation 79 (1988) 43–59.
[16] S. Khuller, B. Schieber, On independent spanning-trees, Information Processing Letters 42 (1992) 321–323.
[17] K. Miura, D. Takahashi, S. Nakano, T. Nishizeki, A linear-time algorithm to find four independent spanning trees in four-connected planar graphs, International Journal of Foundations of Computer Science 10 (1999) 195–210.
[18] K. Miura, D. Takahashi, S. Nakano, T. Nishizeki, A linear-time algorithm to find four independent spanning trees in four-connected planar graphs, Discrete Applied Mathematics 83 (1998) 3–20.

[19] S. Nagai, S. Nakano, A linear-time algorithm to find independent spanning trees in maximal planar graphs, IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences E84A (2001) 1102–1109. Also appears in: Proceedings of 26th Workshop on Graph-Theoretic Concepts in Computer Science, WG 2000, in: LNCS 1928, Springer, 2000, pp. 290–301.
[20] K. Obokata, Y. Iwasaki, F. Bao, Y. Igarashi, Independent spanning trees of product graphs, Lecture Notes in Computer Science 197 (1996) 338–351. See also: K. Obokata, Y. Iwasaki, F. Bao, Y. Igarashi, Independent spanning trees of product graphs and their construction, in: IEICE Trans. Fundamentals of Electronics,Communications and Computer Sciences, E79-A, pp. 1894–1903, 1996.
[21] Y.C. Tseng, S.Y. Wang, C.W. Ho, Efficient broadcasting in wormhole-routed multicomputers: a network-partitioning approach, IEEE Transaction on Parallel and Distributed Systems 10 (1999) 44–61.
[22] S.M. Tang, Y.L. Wang, Y.H. Leu, Optimal independent spanning trees on hypercubes, Journal of Information Science and Engineering 20 (2004) 143–155.
[23] D.B. West, Introduction to Graph Theory, 2nd ed., Prentice Hall, Upper Saddle River, NJ, 2001.
[24] J.S. Yang, J.M. Chang, S.M. Tang, Y.L. Wang, Reducing the height of independent spanning trees in chordal rings, IEEE Transactions on Parallel and Distributed Systems 18 (2007) 644–657.
[25] X. Yang, D.J. Evans, G.M. Megson, The locally twisted cubes, International Journal of Computer Mathematics 82 (2005) 401–413.
[26] J.S. Yang, S.M. Tang, J.M. Chang, Y.L. Wang, Parallel construction of optimal independent spanning trees on hypercubes, Parallel Computing 33 (2007) 73–79.
[27] A. Zwhavi, A. Itai, Three tree-paths, Journal of Graph Theory 13 (1989) 175–188.