Circular Dimensional-Permutations and Reliable Broadcasting for Hypercubes and Möbius Cubes

Baolei Cheng, Jianxi Fan, Jiwen Yang, Xi Wang

Circular Dimensional-Permutations and Reliable Broadcasting for Hypercubes and Möbius cubes

Baolei Cheng[1,2], Jianxi Fan[1,⋆], Jiwen Yang[1], and Xi Wang[1]

School of Computer Science and Technology, Soochow University,
Suzhou 215006, China
{chengbaolei, jxfan, jwyang, 20124027002}@suda.edu.cn
Key Laboratory for Computer Information Processing Technology,
Soochow University, China

Abstract. Reliable broadcasting for interconnection networks can be achieved by constructing multiple independent spanning trees(ISTs) rooted at the same node. In this paper, we prove that there exists $(n-1)!$ sets of ISTs rooted at arbitrary node for Q_n and M_n based on circular dimensional-permutations of 0, 1, ..., $n-1$ and $n \geq 1$. At the same time, we give an parallel algorithm, called BCIST, which is the further study of IST problem for Q_n and M_n in literature. Furthermore, simulation experiments of ISTs based on JUNG framework and different sets of disjoint paths between node 1 and any node $v \in V(0\text{-}M_4)\backslash\{1\}$ for 0-M_4 are also presented.

Keywords: dimensional-permutation, reliable broadcasting, hypercube, Möbius cube, independent spanning tree

1 Introduction

It is well known that hypercubes are widely used in parallel computing systems, which have many advantageous properties such as lower node degree and diameter, higher connectivity, symmetry, and etc [21], [24]. Furthermore, by changing their links between some nodes, the variants of hypercubes, such as Möbius cubes [9], crossed cubes [20], and twisted cubes [1] were proposed, which have better properties [12], [13], [14], [16], [29].

Independent spanning trees(ISTs for short) have been used in reliable broadcasting, secure message distribution [2], reliable communication protocols [17], one-to-all broadcasting [26], the multi-node broadcasting [3], and diagnosis [6]. Therefore, the problem to construct ISTs for a given network is becoming an important issue.

However, there is a well-known conjecture on the existence of ISTs for any network [17][31]:

⋆ Corresponding author.

Conjecture 1. Let G be an n-node-connected network with $n \geq 1$. Then, there exist n node-independent spanning trees rooted at any node for G.

In what follows, we use independent to represent node-independent. For $n \leq 4$, Conjecture 1 was solved [8], [10], [17], [31], but when $n \geq 5$, it has remained open. Consequently, researchers are interested in the study of ISTs for various special networks. Conjecture 1 has been solved for some restricted classes of networks, such as planar networks [15], product networks [23], hypercubes [25], [28], [30], Möbius cubes [5], [6], locally twisted cubes [22], crossed cubes [4], [7], twisted cubes [27], even networks [18], odd networks [19], and etc.

We say that a sequence of n integers is a *permutation* if it contains all integers from 0 to $n-1$ exactly once. Considering the results for hypercubes and Möbius cubes, each paper in literature only considers a set of ISTs for the special network and lacks the discussion of the relation between the permutations and ISTs.

Question 1. Can all permutations of $0, 1, \ldots, n-1$ be used to construct spanning tree and ISTs for the n-dimensional hypercube Q_n and the n-dimensional Möbius cube M_n?

To solve this question, we adopt the definition of circular dimensional-permutation and prove that any circular dimensional-permutation of $0, 1, \ldots, n-1$ can be used to construct n ISTs for Q_n and M_n in this paper, which is the further discussion of spanning trees and ISTs for Q_n and M_n comparing with the results in literature.

The rest of this paper is organized as follows. Section 2 presents some definitions and graph terminologies and notations. Section 3 discusses the IST problem for Q_n and M_n rooted at any node. We draw the conclusion of the paper in the last section.

2 Preliminaries

2.1 Definition of Hypercubes, Möbius Cubes, and ISTs

We use a unique binary string of length n to denote the address of each node in the n-dimensional hypercube Q_n and the n-dimensional Möbius cube M_n. In what follows, nodes and their addresses will be used alternatively. Q_n is a network consists of 2^n nodes. Any two nodes of Q_n are adjacent whenever their corresponding addresses differ in exactly one place.

M_n is a variant of the Q_n, which has two types, 0-*type* n-dimensional Möbius cube and 1-*type* n-dimensional Möbius cube. We adopt the following definition of M_n in [11].

Definition 1. [11] 0-M_1 and 1-M_1 are both the complete graph on two nodes whose addresses are 0 and 1. For any integer n with $n \geq 2$, both 0-M_n and 1-M_n contain one 0-type $(n-1)$-dimensional sub-Möbius cube M^0_{n-1} and one 1-type $(n-1)$-dimensional sub-Möbius cube M^1_{n-1}. The nodes in M^0_{n-1} have a common prefix 0; the nodes in M^1_{n-1} have a common prefix 1. For two nodes $x = x_{n-1}x_{n-2}\ldots x_0 \in V(M^0_{n-1})$ and $y = y_{n-1}y_{n-2}\ldots y_0 \in V(M^1_{n-1})$, where $x_{n-1} = \overline{y_{n-1}} = 0$,

(1) $(x, y) \in E(0\text{-}M_n)$ if and only if $x_i = y_i$, $i = 0, 1, \ldots, n-2$;
(2) $(x, y) \in E(1\text{-}M_n)$ if and only if $x_i = \overline{y_i}$, $i = 0, 1, \ldots, n-2$.

A binary string x of length n is denoted by $x_{n-1}x_{n-2}\ldots x_0$. Suppose that $u = u_{n-1}u_{n-2}\ldots u_0$ and $v = u_{n-1}u_{n-2}\ldots u_l\overline{u_{l-1}}v_{l-2}v_{l-3}\ldots v_0$ are two nodes in $X_n \in \{Q_n, M_n\}$. We say that u and v have *a leftmost differing bit at position* $l-1$. We use LDF(u, v) to denote the leftmost differing bit of two nodes u and v. Given two adjacent nodes u and v, if LDF(u, v)=d, we say that v is the *d-neighbor of* u or that the edge (u, v) is an *edge of dimension* d. For this purpose, let $N_d(u)$ denote the d-neighbor of u. We follow the definitions of *path* and *ancestor* in [7].

Two paths P and P' starting from a node u and ending with another node v are said to be *internally disjoint* if $E(P) \cap E(P') = \varnothing$ and $V(P) \cap V(P') = \{u, v\}$. Two spanning trees for a network G are *independent* if they are rooted at the same node, said u, and for each node $v \in V(G)\backslash\{u\}$, the two paths starting at u and ending with v are internally disjoint. A set of spanning trees of G rooted at v are called *independent spanning trees* if they are pairwisely independent.

2.2 Definition of Dimensional-Permutation

The sequence of n integers is called a *dimensional-permutation* if it contains all integers from 0 to $n-1$ exactly one (Noting that each node in Q_n or M_n has n neighbors, which are 0-neighbor, 1-neighbor, ..., $(n-1)$-neighbor). A *circular dimensional-permutation* (CDP for short) is a type of permutation to put all integers from 0 to $n-1$ along a closed circle in the clockwise order [6]. Suppose that $\{a_0, a_1, \ldots a_{n-1}\}$= {0, 1, ..., $n-1$}. All cyclic permutations of integers are equivalent in the circle,

$a_0 \to a_1 \to \ldots \to a_{n-1}$,
$a_1 \to a_2 \to \ldots \to a_{n-1} \to a_0$,
$\ldots$,
$a_{n-1} \to a_0 \to a_1 \to \ldots \to a_{n-2}$

belong to the same CDP. The total number of CDPs is $n!/n = (n-1)!$. For example, the six CDPs of $0, 1, 2, 3$ are

$3 \to 0 \to 1 \to 2$,
$3 \to 0 \to 2 \to 1$,
$3 \to 1 \to 0 \to 2$,
$3 \to 1 \to 2 \to 0$,
$3 \to 2 \to 0 \to 1$,
$3 \to 2 \to 1 \to 0$.

3 An Reliable Broadcasting Algorithm Based on ISTs for Hypercubes and Möbius Cubes

In this section, we point out that every CDP of n integers $0, 1, 2, \ldots, n-1$ can be used to construct ISTs for $X_n \in \{Q_n, M_n\}$. We now present the following observation.

Observation 1. For the n-dimensional Möbius cube, we proved the correctness of ISTs rooted at any node based on the descending CDP $n-1$, $n-2$, ..., 0 [6]. **In essence, the set of optimal ISTs for Q_n in [25] can be obtained by the ascending CDP 0, 1, ..., $n-1$; the set of ISTs in [30] is similar to that in [28] for Q_n, which can be constructed by the descending CDP $n-1$, $n-2$, ..., 0. Thus, the result in this section is the further study of spanning trees and ISTs for Q_n and M_n.**

3.1 ISTs for Q_n and M_n with any Circular Dimensional-Permutation

In what follows, we always let u denote any node in Q_n or M_n. Now we present an algorithm, called BCIST, to construct n ISTs rooted at an arbitrary node u for $X_n \in \{Q_n, M_n\}$. Fig. 1 demonstrates the construction procedures of n spanning trees T_0, T_1, ..., T_{n-1} rooted at node u for $X_n \in \{Q_n, M_n\}$ in radial style.

Algorithm BCIST
Input: An array $S = \{a_0, a_1, ..., a_{n-1}\}$, where
  the permutation a_0, a_1, ..., a_{n-1} is a CDP of 0, 1, ..., $n-1$;
  an arbitrary node u in $X_n \in \{Q_n, M_n\}$;
Output: $T_0, T_1, ..., T_{n-1}$ rooted at u for X_n;
Begin
for $i = 0$ to $n-1$ do in parallel
  $V(T_i) = N_{a_i}(u)$;
  /*$N_{a_i}(u)$ denotes the a_i-neighbor of u.*/
  $E(T_i) = \varnothing$;
  for $l = 0$ to $n-1$ do
    for each node $v \in V(T_i)$ do in parallel
      $d = S[(i+l+1) \bmod n]$;
      /*Indexing of S is counted from 0 to $n-1$,
      where n is length of the array S.*/
      $E(T_i) = E(T_i) \cup \{(v, N_d(v)\}$;
      $V(T_i) = V(T_i) \cup \{N_d(v)\}$;
    end for
  end for
end for
end

The construction procedures of T_0, T_1, ..., T_{n-1} are similar. Take T_0 for example, the construction procedures are described as follows (See Fig. 1).

At first, there is only one node $N_{a_0}(u)$ in tree T_0; during the 1st iteration (l =0), node $N_{a_0}(u)$ in T_0 is connected to its a_1-neighbor node $N_{a_1}(N_{a_0}(u))$. Therefore, $V(T_0) = \{N_{a_0}(u), N_{a_1}(N_{a_0}(u))\}$ and $E(T_0) = \{(N_{a_0}(u), N_{a_1}(N_{a_0}(u)))\}$; during the 2nd iteration ($l = 1$), each node v in T_0 is connected to its a_2-neighbor node $N_{a_2}(v)$. Thus, the edges $(N_{a_1}(N_{a_0}(u))$, $N_{a_2}(N_{a_1}(N_{a_0}(u))))$ and

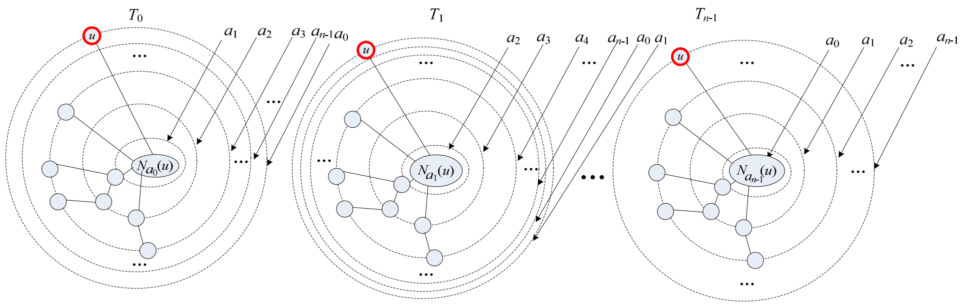

Fig. 1. Construction procedures based on algorithm BCIST.

$(N_{a_0}(u)$, $N_{a_2}(N_{a_0}(u)))$ are appended to T_0 and $V(T_0) = \{N_{a_0}(u), N_{a_2}(N_{a_0}(u)), N_{a_1}(N_{a_0}(u)), N_{a_2}(N_{a_1}(N_{a_0}(u)))\}$; during the 3rd iteration, each node v in T_0 is connected to its a_3-neighbor node $N_{a_3}(v)$. As a result, it has doubled the number of nodes in T_0; During the l-th iteration, each node v in T_0 is connected to its $S[l+1]$-neighbor node $N_{S[l+1]}(v)$ with $4 \leq l \leq n-2$; in the last iteration, each node v in T_0 is connected to its a_0-neighbor node $N_{a_0}(v)$.

Consequently, T_0 is a spanning tree for $X_n \in \{Q_n, M_n\}$.

More examples will be shown in the next subsection. Now we give the following lemma about the relation of adjacent nodes in M_n and Q_n.

Lemma 1. For any node $x_{n-1}x_{n-2}\dots x_0$ and its k-neighbor node $y_{n-1}y_{n-2}\dots y_0$ with $0 \leq k \leq n-1$ in $X_n \in \{M_n, Q_n\}$, we have $x_{n-1}x_{n-2}\dots x_{k+1} = y_{n-1}y_{n-2}\dots y_{k+1}$ and $x_k \neq y_k$.

Based on Definition 1, the definition of Q_n, and Lemma 1, we have the following lemma.

Lemma 2. For any two nodes x, y in $V(X_{n-1}^{x_{n-1}})$, if $\mathrm{LDF}(x,y) = k$ with $0 \leq k \leq n-2$ and $X_{n-1}^{x_{n-1}} \in \{M_{n-1}^{x_{n-1}}, Q_{n-1}^{x_{n-1}}\}$, then $\mathrm{LDF}(N_{n-1}(x), N_{n-1}(y))$=$k$ and $N_{n-1}(x)$, $N_{n-1}(y) \in X_{n-1}^{\overline{x_{n-1}}}$.

Lemma 3. [5] Given a walk W: $u^{(0)}$, $u^{(1)} = N_{m_1}(u^{(0)})$, $u^{(2)} = N_{m_2}(u^{(1)})$, $\dots$, $u^{(k)} = N_{m_k}(u^{(k-1)})$ in M_n for any integer k with $1 \leq k \leq n$, if m_1, m_2, $\dots$, m_k differ from one another and $0 \leq m_i \leq n-1$ for $i = 1, 2, \dots, k$, then W is a path.

For the convenience of proof, we define a vector $< \beta_1, \beta_2, \dots, \beta_n >$ such that the set $\{\beta_1, \beta_2, \dots, \beta_n\}$ equals to the set $\{0, 1, \dots, n-1\}$.

Lemma 4. Suppose that P: $u^{(0)}$, $u^{(1)} = N_{a_1}(u^{(0)})$, $u^{(2)} = N_{a_2}(u^{(1)})$, $\dots$, $u^{(k)} = N_{a_k}(u^{(k-1)})$ and P': $u^{(0)}$, $v^{(1)} = N_{a_1{'}}(u^{(0)})$, $v^{(2)} = N_{a_2{'}}(v^{(1)})$, $\dots$, $v^{(t)} = N_{a_t{'}}(v^{(t-1)})$ are two paths in $X_n \in \{Q_n, M_n\}$ for any two integers k, t with $1 \leq k, t \leq n$ and $a_1 \neq a_1{'}$. If the following conditions hold:
(1) $< \beta_{i_1}, \beta_{i_2}, \dots, \beta_{i_k} >=< a_1, a_2, \dots, a_k >$, where $1 \leq i_1 < i_2 < \dots < i_k \leq n$;
(2) $< \beta_{j_1}, \beta_{j_2}, \dots, \beta_{j_t} >=< a_1{'}, a_2{'}, \dots, a_t{'} >$, where $1 \leq j_1 < j_2 < \dots < j_t \leq n$,
then $V(\mathrm{path}(P, u^{(1)}, u^{(k)})) \cap V(\mathrm{path}(P', v^{(1)}, v^{(t)})) = \varnothing$.

Proof. Suppose that there exists a node v, such that $v \in V(P) \cap V(P')$. We denote v as $v = u^{(i)}$ and $v = v^{(j)}$ with $1 \leq i \leq k$ and $1 \leq j \leq t$. Let $A =$

$\{a_1, a_2, \ldots, a_i\}$ and $B = \{a_1{}', a_2{}', \ldots, a_j{}'\}$. We have $M_1 = \max((A\cup B)\backslash(A\cap B))$. By Lemma 2, we can verify that the M_1-bit of v in P is different from that of v in P', which is a contradiction. □

Based on Lemma 4, we have the following corollary.

Corollary 1. Suppose that P: $u^{(0)}$, $u^{(1)} = N_{a_1}(u^{(0)})$, $u^{(2)} = N_{a_2}(u^{(1)})$, $\ldots$, $u^{(k)} = N_{a_k}(u^{(k-1)}) = v$ and P': $v^{(0)}$, $v^{(1)} = N_{a_1{}'}(u^{(0)})$, $v^{(2)} = N_{a_2{}'}(v^{(1)})$, $\ldots$, $v^{(t)} = N_{a_t{}'}(v^{(t-1)}) = v$ are two paths in $X_n \in \{Q_n, M_n\}$ for any two integers k, t with $1 \le k, t \le n$ and $a_k \ne a_t{}'$. If the following conditions hold:
(1) $<\beta_{i_1}, \beta_{i_2}, \ldots, \beta_{i_k}>=<a_1, a_2, \ldots, a_k>$, where $1 \le i_1 < i_2 < \ldots < i_k \le n$;
(2) $<\beta_{j_1}, \beta_{j_2}, \ldots, \beta_{j_t}>=<a_1{}', a_2{}', \ldots, a_t{}'>$, where $1 \le j_1 < j_2 < \ldots < j_t \le n$,
then $V(\text{path}(P, u^{(0)}, u^{(k)})) \cap V(\text{path}(P', v^{(0)}, v^{(t)})) = \{v\}$.

Lemma 5. Let the input $a_0, a_1, \ldots, a_{n-1}$ of algorithm BCIST be any dimensional-permutation of integers 0, 1,…,$n-1$ and $X_n \in \{Q_n, M_n\}$. T_i obtained by Algorithm BCIST is a spanning tree for X_n with integer $i = 0, 1, \ldots, n-1$.

Proof. Without loss of generality, we consider the tree T_0 obtained by algorithm BCIST. After the n iterations, we have $1+2^0+2^1+...+2^{n-1} = 2^n$ nodes in tree T_0. Choosing arbitrary two nodes $v^{(1)}$ and $v^{(2)}$ from T_0, the $<N_{a_0}(u), v^{(1)}>$-path can be denoted by $N_{a_0}(u)$, $x^{(1)} = N_{a_{i_1}}(N_{a_0}(u))$, $x^{(2)} = N_{a_{i_2}}(x^{(1)})$, $\ldots$, $x^{(k)} = N_{a_{i_k}}(x^{(k-1)})$ and the $<N_{a_0}(u), v^{(2)}>$-path can be denoted by $N_{a_0}(u)$, $x^{(1)} = N_{a_{j_1}}(N_{a_0}(u))$, $x^{(2)} = N_{a_{j_2}}(x^{(1)})$, $\ldots$, $x^{(m)} = N_{a_{j_m}}(x^{(m-1)})$, where $0 \le i_1 < i_2 < \ldots < i_k \le n-1$ and $0 \le j_1 < j_2 < \ldots < j_m \le n-1$. By algorithm BCIST, it is easy to verify $<N_{a_0}(u), v^{(1)}>$-path and $<N_{a_0}(u), v^{(2)}>$-path satisfy the conditions in Lemma 4, which implies that $v^{(1)} \ne v^{(2)}$.

Thus, we can say that T_0 is a spanning tree rooted at $N_{a_0}(u)$ for $X_n \in \{Q_n, M_n\}$. Noting that u is the child of node $N_{a_0}(u)$ and the leaf node in T_0, that is, T_0 is a spanning tree rooted at u. □

Lemma 6. [4] Let T and T' be two spanning trees rooted at node u for a network G. T and T' are independent if and only if for every node $v \in V(G)\backslash\{u\}$, ancestor $(v, T) \cap$ ancestor $(v, T') = \{u\}$ and ancestor $(v, T) \cup$ ancestor $(v, T') \supset \{u\}$.

Lemma 7. $T_0, T_1, \ldots, T_{n-1}$ obtained by Algorithm BCIST are n ISTs for $X_n \in \{Q_n, M_n\}$.

Proof. We have the following two cases.

Case 1. X_n is M_n. By Lemma 5, T_i obtained by Algorithm BCIST is a spanning tree for $V(M_n)$ for integer $i = 0, 1, \ldots, n-1$. We only need to prove that T_i and T_j are independent for $0 \le i \le j \le n-1$.

The trivial cases is $n = 1$ and $n = 2$. Now we consider n with $n \ge 3$. The longest path P_1 in T_i and the longest path P_2 in T_j can be denoted by

P_1: u, $x_0 = N_{a_i}(u)$, $x_1 = N_{a_{i+1}}(x_0)$, $\ldots$, $x_{n-i-1} = N_{a_{n-1}}(x_{n-i-2})$, $x_{n-i} = N_{a_0}(x_{n-i-1})$, $x_{n-i+1} = N_{a_1}(x_{n-i})$, $\ldots$, $x_n = N_{a_i}(x_{n-1})$ and

P_2: u, $y_0 = N_{a_j}(u)$, $y_1 = N_{a_{j+1}}(y_0)$, $\ldots$, $y_{n-j-1} = N_{a_{n-1}}(x_{n-j-2})$, $y_{n-j} = N_{a_0}(y_{n-j-1})$, $y_{n-j+1} = N_{a_1}(y_{n-j})$, $\ldots$, $y_n = N_{a_j}(y_{n-1})$,
respectively, where $0 \le i < j \le n-1$.

Let $a_{c_0} = a_i$ and $a_{d_0} = a_j$. By lemma 6, we only need to prove that for any $v \in V(T_i) \cap V(T_j)$, ancestor $(v, T_i) \cap$ ancestor $(v, T_j) = \{u\}$ and ancestor (v, T_i) $\cup$ ancestor $(v, T_j) \supset \{u\}$. Any path in T_i and any path in T_j can be denoted by P_3 and P_4, respectively, as follows.

P_3: u, $x_0 = N_{a_{c_0}=a_i}(u)$, $x_1{}' = N_{a_{c_1}(x_i)}$, …, $x_k{}' = N_{a_{c_k}}(x_{k-1}{}')$, $x_{k+1}{}' = N_{a_{c_0}}(x_k{}')$ and

P_4: u, $y_0 = N_{a_{d_0}=a_j}(u)$, $y_1{}' = N_{a_{d_1}}(y_i)$, …, $y_l{}' = N_{a_{d_l}}(y_{l-1}{}')$, $y_{l+1}{}' = N_{a_{d_0}}(y_l{}')$,

Without loss of generality, suppose that $a_j > a_i$ and $v \in V(P_3) \cap V(P_4)$. Let $v = a_{c_w} = a_{d_z}$ where $1 \le u \le k$ and $1 \le z \le m$. Based on P_1, P_2, P_3, and P_4, we define walks W_1, W_2, W_3, and W_4 as follows.

W_1: a_i, a_{i+1}, …, a_{n-1}, a_0, a_1, …, a_i,

W_2: a_j, a_{j+1}, …, a_{n-1}, a_0, a_1, …, a_j,

W_3: $a_{c_0} = a_i$, a_{c_1}, …, a_{c_w}, and

W_4: $a_{d_0} = a_j$, a_{d_1}, …, a_{d_z}.

Since $a_{c_0} \ne a_{d_0}$, we have $x_0 \ne y_0$, which implies that ancestor $(v, P_3) \cup$ ancestor $(v, P_4) \supset \{u\}$. We only need to prove that ancestor $(v, P_3) \cap$ ancestor $(v, P_4) = \{u\}$. We have the following Cases.

Case 1.1. $\max(V(W_3)) \ne \max(V(W_4))$. Since $\max(V(W_4)) \ge a_j$, then we have the following subcases.

Case 1.1.1. $\max(V(W_3)) < a_j$ and $\max(V(W_4)) = a_j$. Then, each node in $V(P_4)\backslash\{y_{m+1}{}'\}$ and each node in $V(P_3)$ have a leftmost different bit at position a_j. Then we have $v = y_{m+1}{}'$ and ancestor $(v, P_3) \cap$ ancestor $(v, P_4) = \{u\}$.

Case 1.1.2. $\max(V(W_3 \cup W_4)) > a_j$. Then, the $\max(V(W_3 \cup W_4))$-bit of v in P_3 is different from that of v in P_4. It is a contradiction.

Case 1.2. $\max(V(W_3)) = \max(V(W_4))$. Then, we have the following cases.

Case 1.2.1. $\max(V(W_3)) = \max(V(W_4)) = a_j$. We can verify that $v \ne y_{m+1}{}'$ and $a_j \notin \{a_{d_1}, a_{d_2}, \ldots, a_{d_m}\}$. We can divide path ancestor (v, P_3) into P_{31} and P_{32} as follows.

$P_{31}: u, x_0 = N_{a_{c_0}}(u), x_1{}' = N_{a_{c_1}}(x_0), \ldots, ,x_{f-1}{}' = N_{a_{c_{f-1}}}(x_{f-2}{}')$ and

$P_{32}: x_f{}' = N_{a_{c_f}=a_j}(x_{f-1}{}'), x_{f+1}{}' = N_{a_{c_{f+1}}}(x_f{}'), \ldots, x_w{}' = N_{a_{c_w}}(x_{u-1}{}')$.

By Lemma 2, each node in $V(P_{31})$ and each node in $V(P_4)$ have a leftmost different bit at position a_j. Then, $V(P_4) \cap V(P_{31}) = \varnothing$. By Lemma 2, LDF($x_f{}'$, y_0)= $\max(\{a_{c_0} = a_i, a_{c_1}, \ldots, a_{c(f-1)}\})$. Let $A = \{a_{c_{f+1}}, a_{c_{f+2}}, \ldots, a_{c_w}\}$ and $B = \{a_{d_1}, a_{d_2}, \ldots, a_{d_z}\}$. Furthermore, we have the following cases.

Case 1.2.1.1. $\max((A \cup B)\backslash(A \cap B)) >$ LDF($x_f{}'$, y_0). Then, by Lemma 2, the $\max((A \cup B)\backslash(A \cap B))$-bit of v in P_3 is different from that of v in P_4, which is a contradiction.

Case 1.2.1.2. $\max((A \cup B)\backslash(A \cap B)) =$ LDF($x_f{}'$, y_0). By Algorithm BCIST, T_i and T_j are constructed based on the same CDP. Then, we can verify that $a_{c_w} \ne a_{d_z}$. By Corollary 1, $V(P_4) \cap V(P_{32}) = \{v\}$.

Case 1.2.1.3. $\max((A \cup B)\backslash(A \cap B)) <$ LDF($x_f{}'$, y_0). Then, by Lemma 2, the LDF($x_f{}'$, y_0) bit of v in P_3 is different from that of v in P_4, which is a contradiction.

Case 1.2.2. $\max(V(W_3)) = \max(V(W_4)) = M_1 > a_j$. We can divide path sub-path $< u, v >$-path of P_3 into P_{31} and P_{32} as follows.

$P_{31} : u$, $x_0 = N_{a_{c_0}}(u)$, $x_1{}' = N_{a_{c_1}}(x_0)$, $\ldots$,$x_{f-1}{}' = N_{a_{c_{f-1}}}(x_{f-2}{}')$ and

$P_{32} : x_f{}' = N_{a_{c_f=M_1}}(x_{f-1}{}'), x_{f+1}{}' = N_{a_{c_{f+1}}}(x_f{}')$, $\ldots$, $x_w{}' = N_{a_{c_w}}(x_{u-1}{}')$.

We can divide path sub-path $< u, v >$-path of P_4 into P_{41} and P_{42} as follows.

$P_{41} : u$, $y_j = N_{a_{d_0}=a_j}(u)$, $y_1{}' = N_{a_{d_1}}(y_i)$, $\ldots$, $y_{h-1}{}' = N_{a_{d_{h-1}}}(y_{h-2}{}')$, and

$P_{42} : y_h{}' = N_{a_{d_h}}(y_{h-1}{}')$, $y_{h+1}{}' = N_{a_{d_{h+1}}}(y_h{}')$, $\ldots$, $y_z{}' = N_{a_z}(y_{z-1}{}')$,

By Lemma 2, $V(P_{31}) \cap V(P_{41}) = \{u\}$. Clearly, the M_1-bit of each node in $V(P_{42})$ is different from each node in $V(P_{31})$, thus $V(P_{42}) \cap V(P_{31}) = \varnothing$. Similarly, we have $V(P_{32}) \cap V(P_{41}) = \varnothing$. We only need to prove that $V(P_{32}) \cap V(P_{42}) = \varnothing$. Similarly to Case 1.2.1, we can verify that $a_{c_w} \neq b_{d_z}$. By Corollary 1, $V(P_{32}) \cap V(P_{42}) = \{v\}$.

As a consequence, we have ancestor $(v, P_3) \cap$ ancestor $(v, P_4) = \{u\}$.

Case 2. X_n is Q_n. The proof is similar to that of Case 1.

Based on the above discussion, by Lemma 6, the lemma holds. □

Since there are $(n-1)!$ CDPs of 0, 1, $\ldots$, $n-1$, based on Lemma 7, we have the following theorem.

Theorem 1. Based on Algorithm BCIST, there are $(n-1)!$ sets of ISTs for $X_n \in \{Q_n, M_n\}$.

Comparing with the result in [25], **all the** $(n-1)!$ **sets of ISTs can provide optimal reliable broadcasting for** Q_n. As far as the symmetry is concerned, **all the** $(n-1)!$ **sets of ISTs can also provide optimal reliable broadcasting for** M_n.

3.2 Simulation Experiments of ISTs and Disjoint Paths for 0-M_4

As it is well-known that JUNG framework is a software library which provides a common and extendible language for the modeling, analysis, and visualization of data that can be represented as a network. Using Java multi-thread and JUNG framework, we can easily construct ISTs for Q_n and M_n. For example, Fig. 2 illustrates some construction procedures of four ISTs rooted at 1 for 0-M_4. The four trees can be obtained by four steps with parallel fashion.

Based on different CDPs, we can obtain multiple sets of ISTs. Thus, there are different sets of disjoint paths between arbitrary two nodes. Fig. 3(a)–(f) show six sets of ISTs rooted at 1 for 0-M_4 based on the six CDPs mentioned in Section 2.2. Clearly, there are six sets of n disjoint paths between node 1 and any node $v \in V(0\text{-}M_4)\backslash\{1\}$, among which the total length of four disjoint paths between node 1 and node v may be different. For example, the four disjoint paths between nodes 1 and 13 in Fig. 3(a) are

1→ 0→ 8→ 15→ 12→ 13,
1→ 3→ 2→ 10→ 13,
1→ 5→ 13, and
1→ 9→ 14→ 13.

The four disjoint paths between nodes 1 and 13 in Fig. 3(c) are

(a)
(b)
(c)
(d)

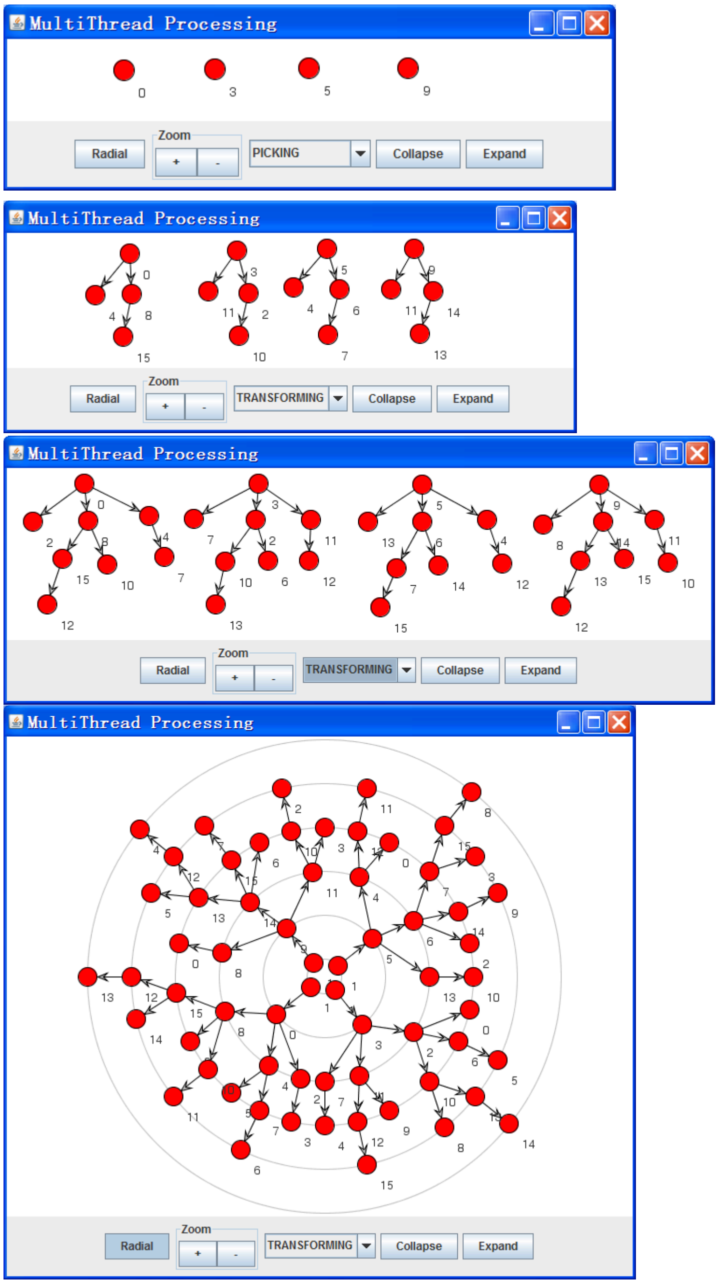

Fig. 2. Simulation experiments of ISTs based on JUNG framework. (a) Initialization; (b) The 2nd iteration; (c) The 3rd iteration; (d) The 4th iteration.

1→ 0→ 2→ 10→ 13,
1→ 3→ 11→ 12→ 13,
1→ 5→ 13, and
1→ 9→ 14→ 13.

The total length of four disjoint paths between nodes 1 and 13 in Fig. 3(c) is shorter than that in Fig. 3(a). Thus, basing on CDPs, we may easily obtain a set of n disjoint paths with better performance between some pairs of nodes.

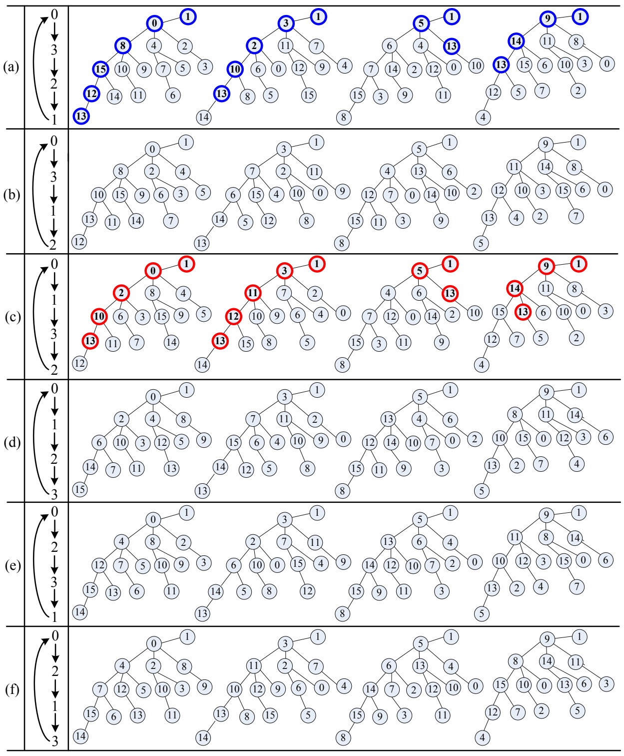

Fig. 3. Example of ISTs rooted at node 1 and disjoint paths between nodes 1 and 13 in 0-M_4.

4 Conclusions

In this paper, we study the circular dimensional-permutations and reliable broadcasting based on ISTs for the n-dimensional hypercube Q_n and the n-dimensional Möbius cube M_n. We prove that any circular dimensional-permutation of 0, 1, $\ldots$, $n-1$ can be used to construct n ISTs, which is the further discussion of IST problem for Q_n and M_n comparing with the results in literature. Moreover, we also conduct simulation experiments of ISTs based on JUNG framework and list some disjoint paths for 0-M_4.

5 Acknowledgment

This work is supported by National Natural Science Foundation of China (No. 61170021), Specialized Research Fund for the Doctoral Program of Higher Education (No. 20103201110018), Application Foundation Research of Suzhou of China (SYG201240), the 2011 Program for Postgraduates Research Innovation in University of Jiangsu Province (No. CXZZ11_0100), the 2012 Science and technology innovation team building program of Soochow University (SDT2012B02) and sponsored by Qing Lan Project.

References

1. Abraham, S., Padmanabhan, K.: The twisted cube topology for multiprocessors: a study in network asymmetry. J. Parallel and Distributed Computing 13(1) (1991) 104–110
2. Bao, F., Igarashi Y., and Öhring, S.R.: Reliable broadcasting in product networks. Discrete Applied Mathematics 83(1–3) (1998) 3–20
3. Chen, Y.-S., Chiang, C.-Y., and Chen, C.-Y.: Multi-node broadcasting in all-ported 3-D wormhole-routed torus using an aggregation-then-distribution strategy. J. Syst. Architect. 50(9) (2004) 575–589
4. Cheng, B., Fan, J., Jia, X., and Zhang, S.: Independent spanning trees in crossed cubes. Information Sciences 233(1) (2013) 276-289
5. Cheng, B., Fan, J., Jia, X., Zhang, S., and Chen, B.: Constructive algorithm of independent spanning trees on Möbius cubes. The Computer Journal (2012) doi: 10.1093/comjnl/bxs123.
6. Cheng, B., Fan, J., Jia, X., and Jia, J.: Parallel construction of independent spanning trees and an application in diagnosis on Möbius Cubes. J. Supercomputing (2013) doi:10.1007/s11227-013-0883-1.
7. Cheng, B., Fan, J., Jia, X., and Wang, J.: Dimension-adjacent trees and parallel construction of independent spanning trees on crossed cubes. J. Parallel and Distributed Computing 73(5) (2013) 641–652
8. Cheriyan, J. and Maheshwari, S.N.: Finding nonseparating induced cycles and independent spanning trees in 3-connected graphs. J. Algorithms 9(4) (1988) 507–537
9. Cull, P. and Larson, S.M.: The Möbius cubes. IEEE Trans. Comput. 44(5) (1995) 647–659
10. Curran, S., Lee, O., and Yu, X.: Finding four independent trees. SIAM J. Comput. 35(5) (2006) 1023–1058
11. Fan J.: Diagnosability of the Möbius Cubes. IEEE Trans. Parallel Distrib. Syst. 9(9) (1998) 923–928
12. Fan, J.: Hamilton-connectivity and cycle-embedding of the Möbius cubes. Inf. Process. Lett. 82(2) (2002) 113–117
13. Fan, J., Jia, X.: Embedding meshes into crossed cubes. Information Sciences 177(15) (2007) 3151–3160
14. Fan, J., Jia, X., Lin, X.: Optimal embeddings of paths with various lengths in twisted cubes. IEEE Trans. Parallel Distrib. Syst. 18(4) (2007) 511–521

15. Huck, A.: Independent trees in planar graphs. Graphs and Combinatorics 15(1) (1999) 29–77
16. Hsieh, S.-Y. and Chen, C.-H.: Pacyclicity on Möbius cubes with maximal edge faults. Parallel Comput. 30(3) (2004) 407–421
17. Itai, A. and Rodeh, M.: The multi-tree approach to reliability in distributed networks. Inform. Comput. 79(1) (1988) 43–59
18. Kim, J.-S., Lee, H.-O., Cheng, E., and Lipták, L.: Independent spanning trees on even networks. Information Sciences 181(13) (2011) 2892–2905
19. Kim, J.-S., Lee, H.-O., Cheng, E., and Lipták, L.: Optimal independent spanning trees on odd graphs. J. Supercomputing 56(2) (2011) 212–225
20. Kulasinghe, P., Bettayeb, S.: Multiply-twisted hypercube with five or more dimensions is not vertex-transitive. Inf. Process. Lett. 53 (1) (1995) 33–36
21. Lee, S.C. and Hook, L.R.: Logic and computer design in nanospace. IEEE Trans. Comput. 57(7) (2008) 965–977
22. Liu, Y.-J., Chou, W.Y., Lan, J.K., and Chen C.: Constructing independent spanning trees for locally twisted cubes. Theoretical Computer Science 412(22) (2011) 2237–2252
23. Obokata, K., Iwasaki, Y., Bao, F., and Igarashi, Y.: Independent spanning trees of product graphs and their construction. IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences E79-A(11) (1996) 1894–1903
24. Schlosser, M., Sintek, M., Decker, S., and Nejdl, W.: HyperCuP—hypercubes, ontologies, and efficient search on peer-to-peer networks. Lecture Notes in Computer Science, Vol. 2530. Springer-Verlag, Berlin Heidelberg, Bologna, Italy (2003) 133–134
25. Tang, S.-M., Wang, Y.-L., and Leu, Y.-H.: Optimal independent spanning trees on hypercubes. J. Information Science and Engineering 20(1) (2004) 143–155
26. Tseng, Y.-C., Wang, S.-Y., and Ho C.-W.: Efficient broadcasting in wormhole-routed multicomputers: A network-partitioning approach. IEEE Trans. Parallel Distrib. Syst. 10(1) (1999) 44–61
27. Wang, Y., Fan, J., Zhou, G., and Jia, X.: Independent spanning trees on twisted cubes. J. Parallel and Distributed Computing 72(1) (2012) 58–69
28. Werapun, J., Intakosum, S., and Boonjing, V.: An efficient parallel construction of optimal independent spanning trees on hypercubes. J. Parallel and Distributed Computing 72(12) (2012) 1713–1724
29. Xu, J.-M., Ma, M., Lü, M.: Paths in Möbius cubes and crossed cubes. Inf. Process. Lett. 97(3) (2006) 94–97
30. Yang, J.-S., Tang, S.-M., Chang, J.-M., and Wang, Y.-L.: Parallel construction of optimal independent spanning trees on hypercubes. Parallel Comput. 33(1) (2007) 73–79
31. Zehavi, A. and Itai, A.: Three tree-paths. J. Graph Theory 13(2) (1989) 175–188