

Independent Spanning Trees on a Multidimensional Torus*

Shyue-Ming Tang^{1†} Jinn-Shyong Yang² Jou-Ming Chang² Yue-Li Wang³

¹ Fu Hsing Kang School, National Defense University, Taipei, Taiwan

² Department of Information Management, National Taipei College of Business, Taipei, Taiwan

³ Department of Computer Science and Information Engineering,
National Chi Nan University, Nantou, Taiwan

Abstract

A set of spanning trees rooted at vertex r in G is called *independent spanning trees* (IST) if for each vertex v in G , $v \neq r$, the paths from v to r in any two trees are different and vertex-disjoint. If the connectivity of G is k , the IST problem is to construct k IST rooted at each vertex. The IST problem has found applications in fault-tolerant broadcasting, but it is still open for general graph with connectivity greater than four. The class of multidimensional tori is a generalization of two-dimensional tori. The class of d -dimensional tori is also a superclass of n -ary d -cubes for $n \geq 2$. Note that hypercubes (binary d -cubes) form a subclass of multidimensional tori. In this paper, we shall propose a simple algorithm to solve the IST problem on a multidimensional torus. Our algorithm is also suitable for parallel or distributed processing.

Keyword: independent spanning trees, internally disjoint paths, parallel algorithm, multidimensional torus, fault-tolerant broadcasting.

*This research is supported by the National Science Council of Taiwan under the Grants NSC93-2416-H-135-001 and NSC95-2115-M-141-001-MY2.

†All correspondence should be addressed to Professor Shyue-Ming Tang, Fu Hsing Kang School, National Defense University, Taipei, Taiwan. (Email: tang@cs.ntust.edu.tw)

1 Introduction

Torus has been proposed as a network topology and widely used in the design of parallel computers. In most cases, the underlying torus networks are two-dimensional. The *multidimensional torus* is a generalization of the two-dimensional torus. A d -dimensional torus, denoted by R_d or $R(m_{d-1}, m_{d-2}, \dots, m_0)$, is a graph consists of $n = \prod_{i=0}^{d-1} m_i$ vertices, where $d \geq 1$ and $m_i \geq 2$ for all i , $0 \leq i \leq d-1$. Each vertex in the torus has a unique label of the form $(a_{d-1}, a_{d-2}, \dots, a_1, a_0)$, where $0 \leq a_i \leq m_i - 1$ for all i , $0 \leq i \leq d-1$. Each vertex $(a_{d-1}, \dots, a_{i+1}, a_i, a_{i-1}, \dots, a_0)$ is connected to all of those vertices labeled by $(a_{d-1}, \dots, a_{i+1}, a_i \pm 1, a_{i-1}, \dots, a_0)$, where $0 \leq i \leq d-1$ and $a_i \pm 1$ is taken modulo m_i . The index i will be referred to as the *dimension* of the edge, while m_i is the *size* of dimension i . As for the definition of a multidimensional torus, see [1, 2, 3, 7]. For example, $R(5, 6)$ is shown in Figure 1. The sizes of dimensions 1 and 0 in $R(5, 6)$ are 5 and 6, respectively.

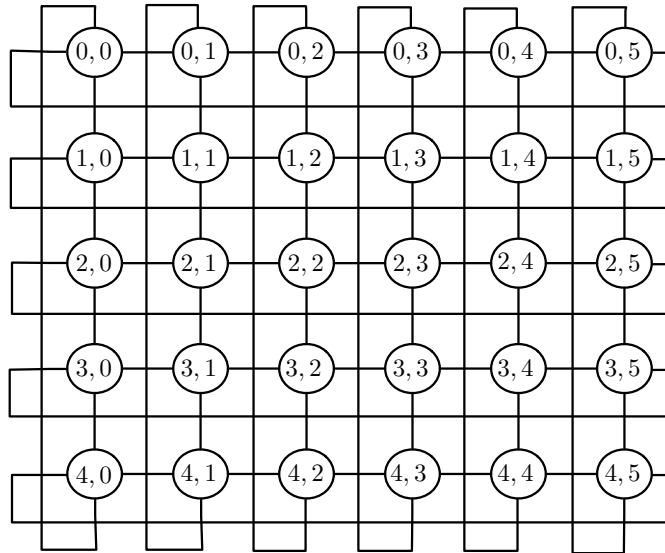


Figure 1: A two-dimensional torus $R(5, 6)$.

As a subclass of product graphs, d -dimensional tori can be defined recursively using the operations of cartesian product, i.e., $R(m_{d-1}, m_{d-2}, \dots, m_0) = C_{m_{d-1}} \times R(m_{d-2}, \dots, m_0)$ and $R(m_0) = C_{m_0}$ [16]. Note that C_{m_i} is either a cycle with m_i vertices or K_2 in case of $m_i = 2$ for $0 \leq i \leq d-1$. By the recursive definition, it turns out that the regularity and connectivity of a d -dimensional torus are in accord with d . We denote by δ the degree of vertices in $R(m_{d-1}, \dots, m_0)$. Then, $\delta = 2\delta_1 + \delta_2$, where δ_1 is the number of $m_i > 2$ and δ_2 is the number of $m_i = 2$. For example, $R(2, 3, 2, 4)$ is a 6-regular graph since $\delta = 2\delta_1 + \delta_2 = 2 \cdot 2 + 2 = 6$.

The class of d -dimensional tori is also a superclass of k -ary d -cubes. In a d -dimensional torus,

different dimensions may have different sizes. Particularly, all dimensions have the same size k in a k -ary d -cube. Note that hypercubes (binary d -cubes) form a subclass of d -dimensional tori.

A set of different paths connecting two vertices in a graph is said to be *internally disjoint* if and only if any two paths in the set have no common vertex except the two end vertices. Considering a graph $G = (V, E)$, a *spanning tree* of G is a subgraph of G that is a tree and contains all vertices in V . Two spanning trees of G are said to be *independent* if they are rooted at the same vertex, say r , and for every vertex $v \in V \setminus \{r\}$, the two paths from r to v , one path in each tree, are internally disjoint. A set of spanning trees on a graph is said to be independent if they are pairwise independent. For example, four independent spanning trees (IST for short) rooted at $(0,0)$ on $R(5,6)$ are shown in Figure 2.

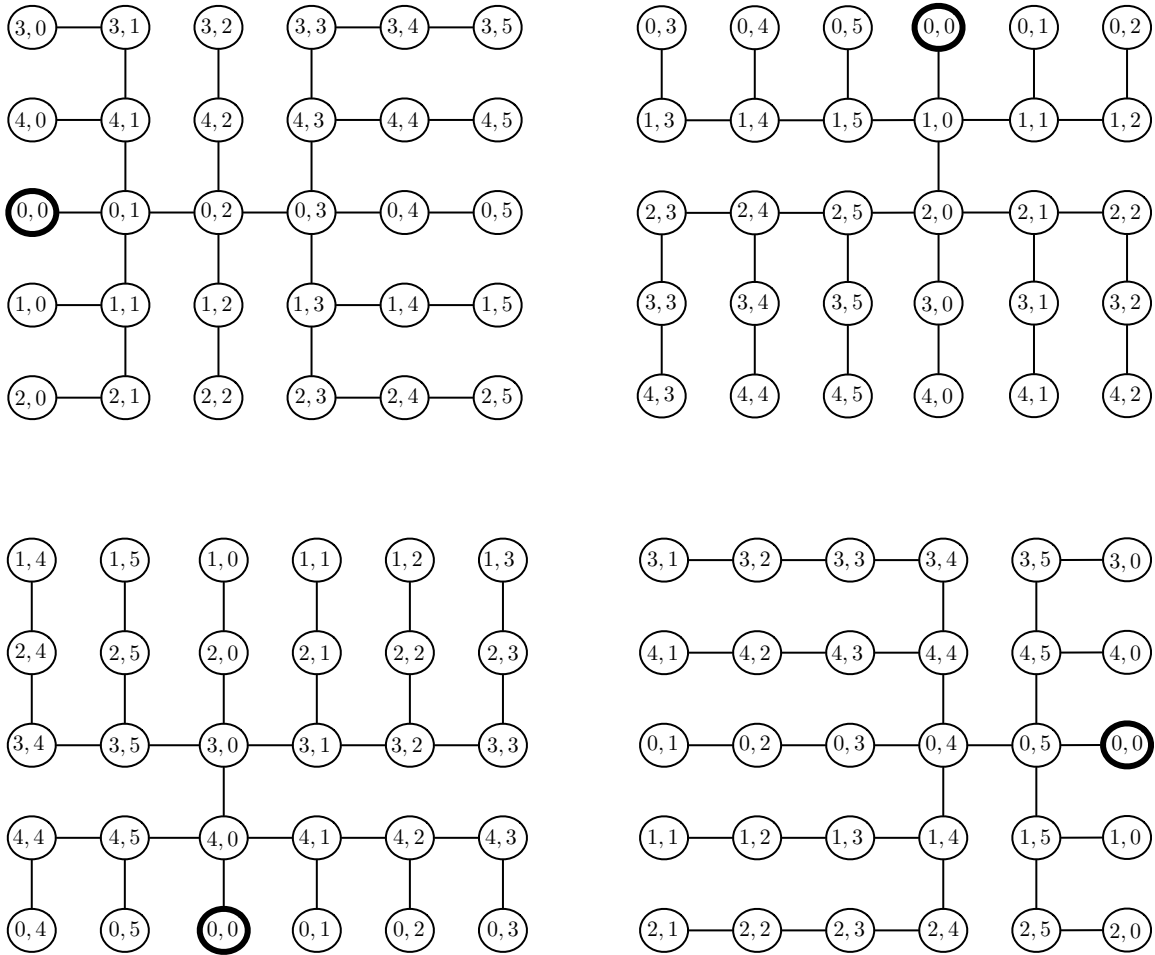


Figure 2: A set of independent spanning trees rooted at vertex $(0,0)$ on $R(5,6)$.

The study of IST problem has contribution to the reliable communication protocols [4, 12]. For example, a rooted spanning tree on an underlying graph can be viewed as a broadcasting channel for data communication. Further, the fault tolerance can be achieved by sending k

copies of the message along the k IST rooted at the source node. If the source node is faultless, this scheme can tolerate up to $k - 1$ faulty nodes and complete the broadcasting process.

Recently, the IST problem has received much attention. However, this is a very hard problem for arbitrary graphs. In fact, Zehavi and Itai [20] conjectured that for any k -connected graph G , there exist k IST rooted at an arbitrary vertex in G . The conjecture has been confirmed only for k -connected graphs with $k \leq 4$ in [5, 6, 12, 20], and it is still open for arbitrary k -connected graphs when $k \geq 5$. Moreover, by providing the construction schemes, the conjecture has been proved to hold for several restricted classes of graphs or digraphs (especially, the graph classes related to interconnection networks), such as planar graphs [11, 15], product graphs [16], chordal rings [13, 18], deBruijn and Kautz digraphs [9, 10], and hypercubes [17, 19], etc. Note that the development of algorithms for constructing IST tends to pursue two research goals, one is to design efficient construction algorithms (e.g., see [13, 15, 18] for linear-time algorithms) and the other is to reduce the height of IST [10, 17, 18].

In [19], Yang et al. proposed a parallel algorithm to construct IST for hypercubes. In this paper, we would like to propose a new algorithm to solve the IST problem on multidimensional tori. Particularly, we will show that our algorithm relies on a simple rule and thus can be implemented easily. The proposed algorithm is also suitable for parallel or distributed processing.

The remaining part of this paper is organized as follows. Section 2 introduces the fundamental concept of the proposed algorithm and gives some essential propositions. Section 3 presents our algorithm that can construct δ IST on a multidimensional torus with degree δ and then shows the correctness of the algorithm. The last section contains our concluding remarks.

2 Preliminaries

Before presenting our algorithm, we have to give some notation and properties which are necessary for illustrating our algorithm. Throughout the paper, we use the term “torus” to mean a multidimensional torus.

To explicitly represent the adjacency of vertices in a torus, we use the notation $x \xrightarrow{j} y$ to mean that vertex y is adjacent to vertex x and j is a *jump* from x to y . We use $\dim(j)$ to stand for the *dimension* of j . Suppose that x_i is the label of the i -th dimension of x and m_i is the size of the i -th dimension of the torus. Then, $x \xrightarrow{j} y$ means that $y_{\dim(j)} \equiv (x_{\dim(j)} \pm 1) \pmod{m_{\dim(j)}}$ and $x_i = y_i$ for $i \neq \dim(j)$, where the operator \pm is in accord with the sign of j . Let J denote the set of all jumps of an arbitrary vertex in a torus. Then, the cardinality of J is the degree of the

torus. If a jump $j \in J$ occurs successively to connect x and y , then we use the notation $x \xrightarrow{\frac{j}{\alpha}} y$ to mean $y_{\dim(j)} \equiv (x_{\dim(j)} \pm \alpha) \pmod{m_{\dim(j)}}$ and $x_i = y_i$ for $i \neq \dim(j)$, where α is the number of repetitions of jump j . Using torus $R(3, 2, 4)$ as an example, we have $(2, 1, 3) \xrightarrow{+0} (2, 1, 0)$, $(2, 1, 3) \xrightarrow{-0} (2, 1, 1)$, $(2, 1, 3) \xrightarrow{+1(-1)} (2, 0, 3)$, $(2, 1, 3) \xrightarrow{+2} (0, 1, 3)$, $(2, 1, 3) \xrightarrow{-2} (0, 1, 3)$, and $J = \{+0, -0, +1(-1), +2, -2\}$.

Since a torus is vertex-symmetric, without loss of generality, we may consider the vertex $(0, 0, \dots, 0)$ (call it vertex 0) as the root of δ IST on a torus. Suppose that \mathcal{T} is a set of δ IST rooted at 0 for R_d . Based on the definition of the IST problem, it is obvious that there is a unique child for the root in every spanning tree in \mathcal{T} . If j is the jump taken from the child to the root, we denote by T_j for the specific spanning tree. For example, the four IST of $R(5, 6)$ shown in Figure 2 are, starting from the top-left one and going clockwise, denoted by T_{-0} , T_{-1} , T_{+0} , and T_{+1} . For each $T \in \mathcal{T}$, the *parent* of a non-root vertex x , denoted by $\text{parent}(T, x)$, is the vertex adjacent to x in the path from x to the root of T .

Suppose that P is a shortest path joining a non-root vertex x and root vertex 0 in R_d . Then, P can be decomposed into a sequence of jumps. Let $x = (x_{d-1}, \dots, x_1, x_0)$ and $R_d = R(m_{d-1}, \dots, m_1, m_0)$. If $0 < x_i \leq \lfloor m_i/2 \rfloor$, a negative jump is used to decrease x_i by one; otherwise, a positive jump is used to increase x_i by one. This process continues until $x_i=0$ for $i = 0, 1, \dots, d-1$. The set of distinct jumps in P is denoted by J_P . If a jump j appears more than once in P , we denote by $n(j)$ the number of occurrences of j . Further, let \bar{j} be a jump with the same dimension and opposite sign of jump j . It is quite obvious that the following properties hold [1, 2]:

- (i) $u \neq v$ for any two vertices $u, v \in P$;
- (ii) $j_i \neq \bar{j}_k$ for any two jumps $j_i, j_k \in J_P$;
- (iii) $n(j) \leq \lfloor \frac{m_{\dim(j)}}{2} \rfloor$ for every $j \in J_P$; and
- (iv) $j, \bar{j} \notin J_P$ if and only if the label of x is 0 in the $\dim(j)$ -th dimension.

A *Latin square* is a square matrix with n^2 entries chosen from a set of n distinct elements such that none of the elements occurs twice within any row or column of the matrix [8]. A *shortest path Latin square* (SPLS for short) with respect to P is a Latin square whose entries are chosen from the jump set J_P . In particular, let $J_P = \{j_0, j_1, \dots, j_{t-1}\}$ with $\dim(j_0) < \dim(j_1) < \dots < \dim(j_{t-1})$. Then, the matrix defined below is called the *increasing SPLS* of P :

$$I(P) = \begin{bmatrix} j_0 & j_1 & \dots & j_{t-2} & j_{t-1} \\ j_1 & j_2 & \dots & j_{t-1} & j_0 \\ \vdots & & \dots & & \\ j_{t-2} & j_{t-1} & \dots & j_{t-4} & j_{t-3} \\ j_{t-1} & j_0 & \dots & j_{t-3} & j_{t-2} \end{bmatrix}.$$

In addition, the jump j_{i-1} in J_P is called the *predecessor* of j_i (denoted by $\text{pred}(J_P, j_i)$), while the jump j_{i+1} is called the *successor* of j_i (denoted by $\text{succ}(J_P, j_i)$). Note that the indices $i-1$ and $i+1$ are taken modulo t . (Hereafter, all arithmetics applied to the indices of the entries in SPLS are taken modulo $|J_P|$.)

Let $r = \sum_{j \in J_P} n(j)$. The *increasing SPLS* of P with repetitions, denoted by $I^*(P)$, is a matrix of size $t \times r$ obtained from $I(P)$ such that every entry j in a row of $I(P)$ is substituted by $n(j)$ repetitions of j in the same row of $I^*(P)$. We can determine a shortest path from x to 0 by taking jumps successively from one row in $I^*(P)$.

Definition 1. Suppose that P is a shortest path from x to 0 in R_d , we denote by $[P]_j$ to mean a shortest path with respect to $I^*(P)$ that contains a specific jump $j \in J_P$ as its last jump. A shortest path of this class is called a *regular path* (r -path).

Particularly, we give the following proposition.

Proposition 2.1. *Let P be a shortest path from x to 0 in R_d . Suppose that two jump sets are taken from left to right, beginning at the left-most jump of distinct rows of $I^*(P)$. The two jump sets are identical if and only if both of them take the whole row of jumps.*

Proof. By the definition of $I^*(P)$, the necessary condition obviously holds. Suppose two jump sets J_1 and J_2 are taken from $I^*(P)$. If J_1 and J_2 are identical, then their corresponding jump sets in distinct rows of $I(P)$ must be identical. Since the jumps are taken from the left-most jump of one row in $I(P)$, the case of identical jump sets never happens unless the whole row of jumps are taken. That means the whole row of jumps are taken in $I^*(P)$. Thus, the efficient condition also holds and the property follows. \square

Based on property (ii), jumps j and \bar{j} cannot simultaneously occur in J_P . We denote by $J_{\bar{P}}$ the set of jumps \bar{j} for all $j \in J_P$. Thus, there is a bijection from J_P to $J_{\bar{P}}$. The following two propositions is fundamental to our algorithm.

Proposition 2.2. *In a torus, let P and Q are two shortest paths from x and y , respectively, to 0. Suppose that $j \in J_P$, $\bar{j} \in J_Q$, and $x \xrightarrow{\bar{j}} y$. If j is positive, then $x_i > \alpha \geq x_i - \lfloor \frac{m_i}{2} \rfloor$ where $i = \text{dim}(j)$. On the other hand, if j is negative, then $m_i - x_i > \alpha \geq \lfloor \frac{m_i}{2} \rfloor + 1 - x_i$.*

Proof. In case of $x_i > \lfloor \frac{m_i}{2} \rfloor$, j is positive and $j \in J_P$ (by property (iii)). Since \bar{j} is negative, x_i can be reduced to a value that is less than or equal to $\lfloor \frac{m_i}{2} \rfloor$ (but greater than 0) after α jumps of \bar{j} from x . That is, $\lfloor \frac{m_i}{2} \rfloor \geq x_i - \alpha > 0$ such that $y_i = x_i - \alpha$ and $\bar{j} \in J_Q$. We rewrite the inequality as $x_i > \alpha \geq x_i - \lfloor \frac{m_i}{2} \rfloor$.

In case of $x_i \leq \lfloor \frac{m_i}{2} \rfloor$, j is negative. Since \bar{j} is positive, x_i can be added to a value that is greater than $\lfloor \frac{m_i}{2} \rfloor$ (but less than m_i) after α jumps of \bar{j} from x . That is, $\lfloor \frac{m_i}{2} \rfloor < x_i - \alpha < m_i$ such that $y_i = x_i + \alpha$. The inequality is rewritten as $m_i - x_i > \alpha \geq \lfloor \frac{m_i}{2} \rfloor + 1 - x_i$. \square

Proposition 2.3. *In a torus, let P and Q are two shortest paths from x and y , respectively, to 0. Both j and \bar{j} do not occur in J_P . If $x \xrightarrow{j} y$, then $\bar{j} \in J_Q$.*

Proof. Let $i = \dim(j)$. Since $j, \bar{j} \notin J_P$, we can infer that $x_i = 0$. After a single jump j from x , x_i will change to a value that is not equal to 0. Thus, $y_i = x_i \pm 1 = 1$ or $m_i - 1$, and \bar{j} must occur in J_Q . \square

As a result, in addition to the class of r -paths, we define two other classes of paths from x to 0. We name them *detour* because they are not shortest paths.

Definition 2. Let x be a non-root vertex in R_d and P be a shortest path from x to 0. If $j \in J_{\bar{P}}$, then there exists a vertex y in R_d such that a *long detour* (ℓ -detour) from x to 0, denoted by $(a \cdot j)|[Q]_j$, is obtained by adding a repetitions of jump j in the front of $[Q]_j$, where Q is a shortest path from y to 0. Let $i = \dim(j)$. In particular, $a = x_i - \lfloor \frac{m_i}{2} \rfloor$ if j is negative; and $a = \lfloor \frac{m_i}{2} \rfloor + 1 - x_i$ if j is positive.

Definition 3. Let x be a non-root vertex in R_d and P be a shortest path from x to 0. If $j \in J \setminus (J_P \cup J_{\bar{P}})$, then there exists a vertex y in R_d such that a *short detour* (s -detour) from x to 0, denoted by $(j)|[Q]_{\bar{j}}$, is obtained by adding j in the front of $[Q]_{\bar{j}}$, where Q is a shortest path from y to 0.

The existence of ℓ -detours and s -detours from x to 0 is inferred directly from Propositions 2.2 and 2.3. In an ℓ -detour, there are a repetitions of j from x to y . Note that a is the minimum value of α that is mentioned in Proposition 2.2. The minimum α is required to ensure that y is the nearest vertex from x such that $j \in J_Q$.

Example 1. We consider torus $R(4, 5, 2, 6, 3)$ and a vertex $x=(2,3,1,0,2)$. Then,

$$P : (2, 3, 1, 0, 2) \xrightarrow{+0} (2, 3, 1, 0, 0) \xrightarrow{-2} (2, 3, 0, 0, 0) \xrightarrow{+3} (2, 0, 0, 0, 0) \xrightarrow{-4} (0, 0, 0, 0, 0)$$

is a shortest path from x to 0. The two particular matrices defined above are given by

$$I(P) = \begin{bmatrix} +0 & -2 & +3 & -4 \\ -2 & +3 & -4 & +0 \\ +3 & -4 & +0 & -2 \\ -4 & +0 & -2 & +3 \end{bmatrix} \quad \text{and} \quad I^*(P) = \begin{bmatrix} +0 & -2 & +3 & +3 & -4 & -4 \\ -2 & +3 & +3 & -4 & -4 & +0 \\ +3 & +3 & -4 & -4 & +0 & -2 \\ -4 & -4 & +0 & -2 & +3 & +3 \end{bmatrix}.$$

The four r -paths with respect to $I^*(P)$ are

$$[P]_{-4} \equiv P,$$

$$\begin{aligned}
[P]_{+0}: & (2,3,1,0,2) \xrightarrow{-2} (2,3,0,0,2) \xrightarrow{\frac{+3}{2}} (2,0,0,0,2) \xrightarrow{\frac{-4}{2}} (0,0,0,0,2) \xrightarrow{+0} (0,0,0,0,0), \\
[P]_{-2}: & (2,3,1,0,2) \xrightarrow{\frac{+3}{2}} (2,0,1,0,2) \xrightarrow{\frac{-4}{2}} (0,0,1,0,2) \xrightarrow{+0} (0,0,1,0,0) \xrightarrow{-2} (0,0,0,0,0), \\
[P]_{+3}: & (2,3,1,0,2) \xrightarrow{\frac{-4}{2}} (0,3,1,0,2) \xrightarrow{+0} (0,3,1,0,0) \xrightarrow{-2} (0,3,0,0,0) \xrightarrow{\frac{+3}{2}} (0,0,0,0,0).
\end{aligned}$$

Let Q_1 be a shortest path from $(2,2,1,0,2)$ to 0 . An r -path with respect to $I^*(Q_1)$ is $[Q_1]_{-3} : (2,2,1,0,2) \xrightarrow{\frac{-4}{2}} (0,2,1,0,2) \xrightarrow{+0} (0,2,1,0,0) \xrightarrow{-2} (0,2,0,0,0) \xrightarrow{\frac{-3}{2}} (0,0,0,0,0)$. Then, $(1 \cdot -3)[Q_1]_{-3}$ forms an ℓ -detour from x to 0 .

Further, let Q_2 be a shortest path from $(2,3,1,1,2)$ to 0 . An r -path with respect to $I^*(Q_2)$ is $[Q_2]_{-1} : (2,3,1,1,2) \xrightarrow{-2} (2,3,0,1,2) \xrightarrow{\frac{+3}{2}} (2,0,0,1,2) \xrightarrow{\frac{-4}{2}} (0,0,0,1,2) \xrightarrow{+0} (0,0,0,1,0) \xrightarrow{-1} (0,0,0,0,0)$. Then, $(+1)[Q_2]_{-1}$ forms an s -detour from x to 0 .

Example 2. We consider torus $R(5,6)$ (as shown in Figure 1) and a shortest path P from $x=(2,5)$ to $(0,0)$. $P : (2,5) \xrightarrow{+0} (2,0) \xrightarrow{\frac{-1}{2}} (0,0)$ and $J_P = \{+0, -1\}$.

Two r -paths with respect to $I^*(P)$ are

$$[P]_{+0}: (2,5) \xrightarrow{\frac{-1}{2}} (0,5) \xrightarrow{+0} (0,0) \text{ and } [P]_{-1}: (2,5) \xrightarrow{+0} (2,0) \xrightarrow{\frac{-1}{2}} (0,0).$$

Let Q_y be a shortest path from $y=(2,3)$ to $(0,0)$. Then, an ℓ -detourh from $(2,5)$ to $(0,0)$ is

$$(2 \cdot -0)[Q_y]_{-0}: (2,5) \xrightarrow{\frac{-0}{2}} (2,3) \xrightarrow{\frac{-1}{2}} (0,3) \xrightarrow{\frac{-0}{3}} (0,0).$$

Further, let Q_z be a shortest path from $z=(3,5)$ to $(0,0)$. Then, another ℓ -detour is

$$(1 \cdot +1)[Q_z]_{+1}: (2,5) \xrightarrow{+1} (3,5) \xrightarrow{+0} (3,0) \xrightarrow{\frac{+1}{2}} (0,0)$$

that also forms a path from $(2,5)$ to $(0,0)$.

In summary, the paths from every vertex to the root vertex in a torus can be divided exclusively into three classes of paths. That is, r -paths, ℓ -detours and s -detours. In Section 4, we will prove that all of the paths are internally disjoint.

3 Constructing Independent Spanning Trees on R_d

In this section, we solve the IST problem on R_d by constructing a set \mathcal{T} of IST rooted at vertex 0 . The construction consists of two phases. In the first phase, we design a procedure for finding a shortest path P from every non-root vertex x to the root in R_d , and then determine the jump set J_P of P . In the second phase, a set of simple rules referred to J_P can be used to determine $\text{parent}(T, x)$ in each tree $T \in \mathcal{T}$.

For each non-root vertex $x = (x_{d-1}, x_{d-2}, \dots, x_0)$ in $R(m_{d-1}, m_{d-2}, \dots, m_0)$, we perform the following procedure to generate the jump set J_P of a shortest path P from vertex x to the root vertex:

Procedure GEN-JUMP-SET(x)
begin

```

 $J_P = \emptyset;$ 
For  $i = 0$  to  $d - 1$  do
  if  $0 < x_i \leq \lfloor m_i/2 \rfloor$ 
     $J_P = J_P \cup \{-i\};$ 
  else  $(\lfloor m_i/2 \rfloor < x_i \leq m_i - 1)$ 
     $J_P = J_P \cup \{+i\};$ 
  endif
enddo
end GEN-JUMP-SET

```

Recall that \bar{j} is a jump with the same dimension and opposite sign of jump j , and $J_{\bar{P}}$ is the set of jumps \bar{j} for all $j \in J_P$. Thus, the cardinality of J_P and $J_{\bar{P}}$ is identical if $m_{\dim(j)} > 2$. In the special case of $m_{\dim(j)} = 2$, we use j for \bar{j} and assign j to be negative.

Let $N_j(x)$ denote the adjacent vertex (or neighbor) of x by taking jump j , i.e., $x \xrightarrow{j} N_j(x)$. To construct an IST set \mathcal{T} on $R(m_{d-1}, m_{d-2}, \dots, m_0)$, we perform the following procedure:

```

Procedure GEN-PARENTS( $x$ )
begin
Step 1. Determine  $J_P$ ,  $J_{\bar{P}}$ , and  $J \setminus (J_P \cup J_{\bar{P}})$  of  $x$  by using Procedure GEN-JUMP-SET( $x$ );
Step 2. for all  $j \in J_P$  do
   $k = \text{pred}(J_P, j);$ 
   $\text{parent}(T_k, x) = N_j(x);$ 
enddo
Step 3. for all  $j \in J_{\bar{P}}$  do
   $\text{parent}(T_j, x) = N_j(x);$ 
enddo
Step 4. for all  $j \in J \setminus (J_P \cup J_{\bar{P}})$  do
   $\text{parent}(T_{\bar{j}}, x) = N_j(x);$ 
enddo
end GEN-PARENTS

```

Example 3. We consider torus $R(4, 5, 2, 6, 3)$ again as an example. Table 1 shows the three types of jumps for $(2, 3, 1, 0, 2)$ and its nine neighbors in the torus. According to the rules in Procedure GEN-PARENTS, information for calculating the parents of vertices $(2, 3, 1, 0, 2)$ and $(2, 3, 0, 0, 2)$ in every spanning tree is summarized in Table 2. From this table, we can see that $(2, 3, 0, 0, 2)$ is the parent of $(2, 3, 1, 0, 2)$ in T_{+0} , while $(2, 3, 1, 0, 2)$ is the parent of $(2, 3, 0, 0, 2)$ in $T_{+2}(T_{-2})$.

In the following section, we will show the correctness of the Procedure GEN-PARENTS.

Table 1: Three types of jumps for vertex $(2, 3, 1, 0, 2)$ and its neighbors in $R(4, 5, 2, 6, 3)$

x	J_P	$J_{\bar{P}}$	$J \setminus (J_P \cup J_{\bar{P}})$
$(2, 3, 1, 0, 2)$	$\{+0, -2(+2), +3, -4\}$	$\{-0, -3, +4\}$	$\{+1, -1\}$
$(2, 3, 1, 0, 0)$	$\{-2(+2), +3, -4\}$	$\{-3, +4\}$	$\{+0, -0, +1, -1\}$
$(2, 3, 1, 0, 1)$	$\{-0, -2(+2), +3, -4\}$	$\{+0, -3, +4\}$	$\{+1, -1\}$
$(2, 3, 1, 1, 2)$	$\{+0, -1, -2(+2), +3, -4\}$	$\{-0, +1, -3, +4\}$	\emptyset
$(2, 3, 1, 5, 2)$	$\{+0, +1, -2(-2), +3, -4\}$	$\{-0, -1, -3, +4\}$	\emptyset
$(2, 3, 0, 0, 2)$	$\{+0, +3, -4\}$	$\{-0, -3, +4\}$	$\{+1, -1, -2(+2)\}$
$(2, 4, 1, 0, 2)$	$\{+0, -2(+2), +3, -4\}$	$\{-0, -3, +4\}$	$\{+1, -1\}$
$(2, 2, 1, 0, 2)$	$\{+0, -2(+2), -3, -4\}$	$\{-0, +3, +4\}$	$\{+1, -1\}$
$(3, 3, 1, 0, 2)$	$\{+0, -2(+2), +3, +4\}$	$\{-0, -3, -4\}$	$\{+1, -1\}$
$(1, 3, 1, 0, 2)$	$\{+0, -2(+2), +3, -4\}$	$\{-0, -3, +4\}$	$\{+1, -1\}$

Table 2: The parents of vertices $(2, 3, 1, 0, 2)$ and $(2, 3, 0, 0, 2)$ in each IST of $R(4, 5, 2, 6, 3)$.

x	the condition of jump j	IST	parent of x
$(2, 3, 1, 0, 2)$	$j = -0 \in J_{\bar{P}}$	T_0^-	$(2, 3, 1, 0, 1)$
	$j = +0 \in J_P$	T_4^-	$(2, 3, 1, 0, 0)$
	$j = -1 \in J \setminus (J_P \cup J_{\bar{P}})$	T_1^+	$(2, 3, 1, 5, 2)$
	$j = +1 \in J \setminus (J_P \cup J_{\bar{P}})$	T_1^-	$(2, 3, 1, 1, 2)$
	$j = -2(+2) \in J_P$	T_0^+	$(2, 3, 0, 0, 2)$
	$j = -3 \in J_{\bar{P}}$	T_3^-	$(2, 2, 1, 0, 2)$
	$j = +3 \in J_P$	$T_2^-(T_2^+)$	$(2, 4, 1, 0, 2)$
	$j = -4 \in J_P$	T_3^+	$(1, 3, 1, 0, 2)$
	$j = +4 \in J_{\bar{P}}$	T_4^+	$(3, 3, 1, 0, 2)$
	$(2, 3, 0, 0, 2)$	$j = -0 \in J_{\bar{P}}$	T_0^-
$j = +0 \in J_P$		T_4^-	$(2, 3, 0, 0, 0)$
$j = -1 \in J \setminus (J_P \cup J_{\bar{P}})$		T_1^+	$(2, 3, 0, 5, 2)$
$j = +1 \in J \setminus (J_P \cup J_{\bar{P}})$		T_1^-	$(2, 3, 0, 1, 2)$
$j = -2(+2) \in J \setminus (J_P \cup J_{\bar{P}})$		$T_2^+(T_2^-)$	$(2, 3, 1, 0, 2)$
$j = -3 \in J_{\bar{P}}$		T_3^-	$(2, 2, 0, 0, 2)$
$j = +3 \in J_P$		T_0^+	$(2, 4, 0, 0, 2)$
$j = -4 \in J_P$		T_3^+	$(1, 3, 0, 0, 2)$
$j = +4 \in J_{\bar{P}}$		T_4^+	$(3, 3, 0, 0, 2)$

4 Correctness

To show the correctness of Procedure GEN-PARENTS, we have to prove that all of the subgraphs generated by the procedure are spanning trees, and that the unique path from every non-root vertex to the root in each spanning tree is internally disjoint with its paths in other spanning trees. Lemma 4.1 gives the proof of the spanning trees.

Lemma 4.1. *Procedure GEN-PARENTS can generate a set of δ spanning trees rooted at vertex 0 in R_d , where δ is the degree of R_d .*

Proof. In Procedure GEN-PARENTS, the cardinality of the jump set J is δ , and every non-root vertex x takes different jump to get its parent in δ different graph (not yet proved to be a tree). It turns out that every resulting graph is a spanning subgraph of R_d . To complete this proof,

we need to show that there exists a unique path from every vertex x to 0 in each subgraph. Let P be a shortest path from x to 0 in R_d . According to the class of jump j with respect to P , we consider the following three cases:

Case 1: $j \in J_P$. According Step 2 of Procedure GEN-PARENTS, j is the first jump of an r -path (i.e., $[P]_k$, where $k = \text{pred}(J_P, j)$). Since all other internal vertices in $[P]_k$ also take part of the path as their r -paths, the path is unique.

Case 2: $j \in J_{\bar{P}}$. Let $i = \text{dim}(j)$. According Step 3, j is the first jump of an ℓ -detour. That is, there exists a vertex y in the torus, such that $x \xrightarrow{j} y$ (a has been defined in Definition 2) and $j \in J_Q$ (Q is a shortest path from y to 0). Path $(a \cdot j)|[Q]_j$ is an ℓ -detour of x . Every vertex between x and y also takes part of the path as its ℓ -detour. For every internal vertex in $[Q]_j$, it takes part of $[Q]_j$ as its r -path. Thus, $(a \cdot j)|[Q]_j$ forms a unique path from x to 0 in the spanning subgraph.

Case 3: $j \in J \setminus (J_P \cup J_{\bar{P}})$. According Step 4, j leads an s -detour from x to 0. That is, there exists a vertex y in the torus, such that $x \xrightarrow{j} y$ and $\bar{j} \in J_Q$ (Q is a shortest path from y to 0). Path $(j)|[Q]_{\bar{j}}$ is an s -detour of x . For every internal vertex in $[Q]_{\bar{j}}$, it takes part of $[Q]_{\bar{j}}$ as its r -path. As a result, $(j)|[Q]_{\bar{j}}$ forms a unique path from x to 0 in the spanning subgraph.

Consequently, the resulting δ spanning subgraphs of Procedure GEN-PARENTS are spanning trees of R_d . □

We now show the independency of IST in \mathcal{T} .

Lemma 4.2. *Let P be a shortest path from x to 0 in a torus. Then, any two distinct r -paths with respect to $I^*(P)$ are internally disjoint.*

Proof. Suppose that $j_1, j_2 \in J_P$, $[P]_{j_1}$ and $[P]_{j_2}$ are two distinct r -paths from x to 0. Let u and v be any internal vertices in $[P]_{j_1}$ and $[P]_{j_2}$, respectively. If $u = v$, then u and v must have taken the same set of jumps from x to their current positions. According to Proposition 2.1, the case of identical jump sets never happens when a and b are both internal vertices. Thus, the lemma follows. □

Lemma 4.3. *If x is a vertex in a torus, then an r -path and an ℓ -detour from x to 0 are internally disjoint.*

Proof. Suppose that P and Q are two shortest paths from x and y , respectively, to 0. Let $[P]_{j_x}$ be an r -path and $(a \cdot j_y)|[Q]_{j_y}$ be an ℓ -detour from x to 0. Let $i = \text{dim}(j_y)$. By Definitions

1 and 2, we can infer that $\overline{j_y} \in J_P$, $j_y \in J_Q$, and $x \xrightarrow[a]{j_y} y$, where

$$a = \begin{cases} x_i - \lfloor \frac{m_i}{2} \rfloor & \text{if } j_y \text{ is negative,} \\ \lfloor \frac{m_i}{2} \rfloor + 1 - x_i & \text{if } j_y \text{ is positive.} \end{cases}$$

Let u and v be two internal vertices in $[P]_{j_x}$ and $(a \cdot j_y)|[Q]_{j_y}$, respectively. If j_y is positive, then $0 \leq u_i \leq \lfloor \frac{m_i}{2} \rfloor$ but $m_i > v_i > \lfloor \frac{m_i}{2} \rfloor$. That is, $u_i < v_i$. If j_y is negative, the roles of u_i and v_i in the inequalities are exchanged. Thus, $u \neq v$ and the lemma follows. \square

Lemma 4.4. *If x is a vertex in a torus, then an r -path and an s -detour from x to 0 are internally disjoint.*

Proof. Suppose that P and Q are two shortest paths from x and y , respectively, to 0. Let $[P]_{j_x}$ be an r -path and $(\overline{j_y})|[Q]_{j_y}$ be an s -detour from x to 0. By Definitions 1 and 3, we can infer that $j_y, \overline{j_y} \notin J_P$, $j_y \in H_Q$, and $x \xrightarrow{\overline{j_y}} y$.

Let u and v be two internal vertices in $[P]_{j_x}$ and $(\overline{j_y})|[Q]_{j_y}$, respectively. Let $i = \dim(j_y)$. Then, $u_i = x_i = 0$. However, $v_i \neq 0$ since $j_y \in H_Q$ and j_y is the last jump in the path from y to 0. This completes the proof. \square

Lemma 4.5. *If x is a vertex in a torus, then any two distinct ℓ -detours from x to 0 are internally disjoint.*

Proof. Suppose that P and Q are two shortest paths from vertices y and z , respectively, to vertex 0 such that $j_y \in J_P$ and $j_z \in J_Q$. Further, $(a \cdot j_y)|[P]_{j_y}$ and $(b \cdot j_z)|[Q]_{j_z}$ forms two ℓ -detours from x to 0, where a and b are given as Definition 2.

Let u and v be any internal vertices in $(a \cdot j_y)|[P]_{j_y}$ and $(b \cdot j_z)|[Q]_{j_z}$, respectively. Let $i = \dim(j_y)$. By Lemma 4.3, for any internal vertex w in an r -path from x to 0, $w_i \neq u_i$. We can infer that $v_i \neq u_i$ since v and w have the same label range in the i -th dimension. Thus, $u \neq v$ and the lemma follows. \square

Lemma 4.6. *If x is a vertex in a torus, then an ℓ -detour and an s -detour from x to 0 are internally disjoint.*

Proof. Suppose that P and Q are two shortest paths from vertices y and z , respectively, to vertex 0 such that $j_y \in J_P$ and $j_z \in J_Q$. Further, $(a \cdot j_y)|[P]_{j_y}$ is an ℓ -detour and $(\overline{j_z})|[Q]_{j_z}$ is an s -detours from x to 0.

Let u and v be any internal vertices in $(a \cdot j_y)|[P]_{j_y}$ and $(\overline{j_z})|[Q]_{j_z}$, respectively. Let $i = \dim(j_y)$. By Lemma 4.3, for any internal vertex w in an r -path from x to 0, $w_i \neq u_i$. We can infer that

$v_i \neq u_i$ since v and w have the same label range in the i -th dimension. Thus, $u \neq v$ and the lemma follows. \square

Lemma 4.7. *If x is a vertex in a torus, then any two distinct s -detours from x to 0 are internally disjoint.*

Proof. Suppose that P and Q are two shortest paths from vertices y and z , respectively, to vertex 0 such that $j_y \in J_P$ and $j_z \in J_Q$. Further, $(\overline{j_y})|[P]_{j_y}$ and $(\overline{j_z})|[Q]_{j_z}$ are two s -detours from x to 0.

Let u and v be any internal vertices in $(\overline{j_y})|[P]_{j_y}$ and $(\overline{j_z})|[Q]_{j_z}$, respectively. Let $i = \dim(j_y)$. By Lemma 4.4, for any internal vertex w in an r -path from x to 0, $w_i \neq u_i$. We can infer that $v_i \neq u_i$ since v and w have the same label range in the i -th dimension. Thus, $u \neq v$ and the lemma follows. \square

By combining Lemmas 4.1, 4.2, 4.3, 4.4, 4.5, 4.6 and 4.7, we give the following theorem.

Theorem 1. *The algorithm of constructing δ IST on R_d can be completed in $O(\delta n)$ time, where δ is the degree of R_d and n is the number of vertices in R_d . In particular, the algorithm can be parallelized on R_d to run in $O(\delta)$ time.*

5 Concluding Remarks

In this paper, we have presented a simple algorithm for solving the IST problem on multidimensional tori. The algorithm can be parallelized on a torus multiprocessor to run in $O(\delta)$ time, where δ is the degree of R_d .

In [16], Obokata et al. have proved that the IST problem can be solved on a product graph of two graphs on which the Zehavi and Itai's conjecture holds. According to their proof, a d -dimensional torus R_d can be viewed as the product graph of a cycle C and an R_{d-1} . Then, the IST on R_d can be constructed recursively from the IST on C and R_{d-1} . However, their construction algorithm is recursive and forbids the possibility of parallel processing. Our algorithm compensates for the defects of Obokata's algorithm.

Since the root has only one child in every IST, an IST is *optimal* if the path from every vertex to the root-child vertex is a shortest path [17]. In [19], Yang et al. have designed a parallel algorithm for generating k optimal IST on a k -dimensional hypercube. We solve the IST problem on a superclass of hypercubes. The generated IST is optimal if R_d is a hypercube. In other cases, the generated IST might not be optimal. An explanation is given here.

Let P be a path from x to 0 in an IST generated by our algorithm. Let Q be the path from x to the root-child vertex. Then, Q must be a shortest path if P is an r -path or an s -detour. In case that P is an ℓ -detour, however, Q is a shortest path if and only if property (iii) holds. That is, the repetitions of a jump j with respect to Q must be less than or equal to $\lfloor \frac{m_{\dim(j)}}{2} \rfloor$.

Finally, we shall point out that the constructing scheme of IST on a torus is not unique. For instance, if the *increasing* SPLS mentioned in Section 2 is replaced by a *decreasing* SPLS, a different set of IST can be obtained. The enumeration of IST sets on a torus is another interesting issue for future study.

References

- [1] B. F. A. AlMohammad, and B. Bose, Fault-tolerant communication algorithms in toroidal networks, *IEEE Transactions on Parallel and Distributed Systems*, 10(10) (1999), pp. 976–983.
- [2] L. N. Bhuyan, and D. P. Agrawal, Generalized hypercube and hyperbus structures for a computer network, *IEEE Transactions on Computers*, 33(4) (1984), pp. 323–333.
- [3] M. M. Bae, and B. Bose, Resource placement in torus-based networks, *IEEE Transactions on Computers*, 46(10) (1997), pp. 1083–1092.
- [4] F. Bao, Y. Igarashi, and S.R. Öhring, Reliable broadcasting in product networks, *Discrete Applied Mathematics*, 83 (1998), pp. 3–20.
- [5] J. Cheriyan and S.N. Maheshwari, Finding nonseparating induced cycles and independent spanning trees in 3-connected graphs, *Journal of Algorithms*, 9 (1988), pp. 507–537.
- [6] S. Curran, O. Lee, and X. Yu, Finding four independent trees, *SIAM Journal on Computing*, 35 (2006), pp. 1023–1058.
- [7] R. Cypher, and L. Gravano, Storage-efficient, deadlock-free packet routing algorithms for torus networks, *IEEE Transactions on Computers*, 43(12) (1994), pp. 1376–1385.
- [8] J. Dénes, and A. D. Keedwell, *Latin squares and their applications*, Academic Press, New York, 1974.
- [9] Z. Ge and S.L. Hakimi, Disjoint rooted spanning trees with small depths in de Bruijn and Kautz graphs, *SIAM Journal on Computing*, 26 (1997), pp. 79–92.
- [10] T. Hasunuma and H. Nagamochi, Independent spanning trees with small depths in iterated line digraphs, *Discrete Applied Mathematics*, 110 (2001), pp. 189–211.
- [11] A. Huck, Independent trees in planar graphs, *Graphs and Combinatorics*, 15 (1999), pp. 29–77.

- [12] A. Itai and M. Rodeh, The multi-tree approach to reliability in distributed networks, *Information and Computation*, 79 (1988), pp. 43–59.
- [13] Y. Iwasaki, Y. Kajiwara, K. Obokata, and Y. Igarashi, Independent spanning trees of chordal rings, *Information Processing Letters*, 69 (1999), pp. 155–160.
- [14] S.L. Johnsson and C.-T. Ho, Optimum broadcasting and personalized communication in hypercube, *IEEE Transactions on Computers*, 38(9) (1989), pp. 1249–1268.
- [15] K. Miura, D. Takahashi, S. Nakano, and 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), pp. 195–210.
- [16] K. Obokata, Y. Iwasaki, F. Bao, and Y. Igarashi, Independent spanning trees of product graphs and their construction, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E79-A (1996), pp. 1894–1903.
- [17] S.-M. Tang, Y.-L. Wang, and Y.-H. Leu, Optimal independent spanning trees on hypercubes, *Journal of Information Science and Engineering*, 20 (2004), pp. 605–617.
- [18] J.-S. Yang, J.-M. Chang, S.-M. Tang, and Y.-L. Wang, Reducing the height of independent spanning trees in chordal rings, *IEEE Transactions on Parallel and Distributed Systems*, 18(5) (2007), pp.644–657.
- [19] J.-S. Yang, S.-M. Tang, J.-M. Chang, and Y.-L. Wang, Parallel construction of independent spanning trees on hypercubes, *Parallel Computing*, 33 (2007), pp.73-79.
- [20] A. Zehavi and A. Itai, Three tree-paths, *Journal of Graph Theory*, 13 (1989), pp. 175–188.