ARTICLE IN PRESS
Discrete Applied Mathematics ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

A parallel algorithm for constructing independent spanning trees in twisted cubes[☆]

Jou-Ming Chang[a,*], Ting-Jyun Yang[a,b], Jinn-Shyong Yang[c]

[a] Institute of Information and Decision Sciences, National Taipei University of Business, Taipei, Taiwan
[b] Nippon Electric Company, Taiwan
[c] Department of Information Management, National Taipei University of Business, Taipei, Taiwan

ARTICLE INFO

Article history:
Received 3 April 2016
Received in revised form 6 September 2016
Accepted 6 November 2016
Available online xxxx

Keywords:
Parallel algorithms
Independent spanning trees
Interconnection networks
Twisted cubes

ABSTRACT

A long-standing conjecture mentions that a k-connected graph G admits k independent spanning trees (ISTs for short) rooted at an arbitrary node of G. An n-dimensional twisted cube, denoted by TQ_n, is a variation of hypercube with connectivity n and has many features superior to those of hypercube. Yang (2010) first proposed an algorithm to construct n edge-disjoint spanning trees in TQ_n for any odd integer $n \geqslant 3$ and showed that half of them are ISTs. At a later stage, Wang et al. (2012) inferred that the above conjecture in affirmative for TQ_n by providing an $\mathcal{O}(N \log N)$ time algorithm to construct n ISTs, where $N = 2^n$ is the number of nodes in TQ_n. However, this algorithm is executed in a recursive fashion and thus is hard to be parallelized. In this paper, we revisit the problem of constructing ISTs in twisted cubes and present a non-recursive algorithm. Our approach can be fully parallelized to make the use of all nodes of TQ_n as processors for computation in such a way that each node can determine its parent in all spanning trees directly by referring its address and tree indices in $\mathcal{O}(\log N)$ time.

1. Introduction

Let $V(G)$ and $E(G)$ denote the node set and edge set of a graph G, respectively. A set of spanning trees in G is called *independent spanning trees* (ISTs for short) if all the trees are rooted at the same node r such that, for any other node $v \in V(G) \setminus \{r\}$, the paths from v to r in any two trees are internally node-disjoint (i.e., there exists no common node in the two paths except v and r). Constructing multiple spanning trees in a graph has been studied not only from a theoretical point of view but also from some practical applications. In particular, the construction of ISTs is important due to the applications to fault-tolerant broadcasting and secure message distribution in reliable interconnection networks [2,13,18,22]. Suppose that we have k ISTs rooted at a node r in a network. For the former, the fault-tolerant broadcasting can be achieved by sending k copies of a message along k ISTs on the network provided that there are at most $k - 1$ faulty nodes (different from r) and/or faulty edges in the network. For the latter, if a message at the source node is separated into k different parts, by sending the k parts along k ISTs on the network, then the message is secure in the message distributing.

☆ This research was partially supported by Grants MOST104-2221-E-141-002-MY3 (Jou-Ming Chang) and MOST104-2221-E-141-004 (Jinn-Shyong Yang) from the Ministry of Science and Technology, Taiwan.
* Corresponding author.
E-mail address: spade@ntub.edu.tw (J.-M. Chang).

http://dx.doi.org/10.1016/j.dam.2016.11.017
0166-218X/

Let $G - F$ stand for the graph obtained from G by removing a subset $F \subset V(G)$. A graph G is *k-connected* if $|V(G)| > k$ and $G - F$ is connected for any subset $F \subset V(G)$ with $|F| < k$. Zehavi and Itai [29] proposed the following conjecture: if G is a k-connected graph, then it admits k ISTs rooted at an arbitrary node. Till now, the conjecture has been affirmed only for $k \leqslant 4$ (see [13,6,29,7] for $k = 2, 3, 4$, respectively) and is still open for $k \geqslant 5$. Also, research of this topic for applications has been focused on some interconnection networks (see [4,5,12,18,19,22–24,26,28,27] and quotes therein).

Hilbers et al. [11] first introduced the family of twisted cubes as a variation of hypercubes in order to achieving some improvements of structure properties in contrast to hypercubes. Let TQ_n denote the n-dimensional twisted cube. Chang et al. [3] showed that TQ_n is an n-connected graph with the diameter, wide diameter, and faulty diameter being about half of those in comparable hypercube. Although Abraham and Padmanabhan [1] pointed out the asymmetry of twisted cubes, this does not diminish an upsurge of research on twisted cubes. A lot of research results on TQ_n can be found in the literature [8–10,16,14,15,17,19,21,25,20]. In particular, Yang [21] proposed an algorithm to construct n edge-disjoint spanning trees in TQ_n for any odd integer $n \geqslant 3$ and showed that half of them are ISTs. Wang et al. [19] further proposed an algorithm for constructing n ISTs with an arbitrary node as the root in $\mathcal{O}(N \log N)$ time, where $N = 2^n$ is the number of nodes in TQ_n. However, this algorithm is executed in a recursive fashion and thus is hard to be parallelized.

In this paper, we revisit the problem of constructing ISTs in twisted cubes. An algorithm developed for an interconnection network is said to be *fully parallelized* if it could make the use of all nodes of such a network as processors for computation. An obvious advantage of a fully parallelized algorithm is that it can lead to a better utilization of computing resources. Accordingly, we present a non-recursive and fully parallelized algorithm for constructing n ISTs rooted at an arbitrary node in TQ_n. A crucial practice of our algorithm is to find the parent of every node in each spanning tree directly, and consequently the algorithm can be parallelized to run in $\mathcal{O}(\log N)$ time using $N = 2^n$ nodes of TQ_n as processors. Note that the computation of each node only relies on the information of its address and tree indices. The algorithm is efficient because there are n ISTs to be constructed and each IST contains 2^n nodes.

The rest of this paper is organized as follows. Section 2 formally gives the definition of twisted cubes and some useful terminologies and notations. Section 3 presents our algorithm for constructing n ISTs in TQ_n and provides examples as illustration. Section 4 proves the correctness of the algorithm and analyzes its complexity. The final section contains our concluding remarks.

2. Preliminary

Let $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$. For a binary string $x = x_{n-1}x_{n-2}\cdots x_0$ and an integer $i \in \mathbb{Z}_n$, we define $\dddot{\oplus}(x, i) = x_i \oplus x_{i-1} \oplus \cdots \oplus x_0$, where $\oplus$ is the bit exclusive-OR operation. The n-dimensional twisted cube, denoted by TQ_n, is a variant of the n-dimensional hypercube with 2^n nodes, where each node is labeled by a unique binary string of length n as its address. TQ_n can be recursively defined as follows.

Definition 1 (*[11]*)**.** The 1-dimensional twisted cube TQ_1 is the complete graph with two nodes labeled by 0 and 1. For an odd integer $n \geqslant 3$, TQ_n consists of four subcubes TQ^{00}_{n-2}, TQ^{01}_{n-2}, TQ^{10}_{n-2}, and TQ^{11}_{n-2}, where TQ^{ab}_{n-2} for $a, b \in \mathbb{Z}_2$ is isomorphic to TQ_{n-2} such that $V(TQ^{ab}_{n-2}) = \{abx\colon\ x \in V(TQ_{n-2})\}$ (i.e., adding two preceding bits a and b in the front of a node labeled by x) and $E(TQ^{ab}_{n-2}) = \{(abx, aby)\colon\ (x, y) \in E(TQ_{n-2})\}$. That is, $V(TQ_n) = \cup_{ab\in\mathbb{Z}_2} V(TQ^{ab}_{n-2})$. Define $E(TQ_n) = \cup_{ab\in\mathbb{Z}_2} E(TQ^{ab}_{n-2}) \cup E'$, where an edge $(u, v) \in E'$ if and only if the two nodes $u = u_{n-1}u_{n-2}\cdots u_0$ and $v = v_{n-1}v_{n-2}\cdots v_0$ satisfy one of the following conditions:

(1) $u = \bar{v}_{n-1}v_{n-2}\cdots v_0$;
(2) $u = \bar{v}_{n-1}\bar{v}_{n-2}v_{n-3}\cdots v_0$ for $\dddot{\oplus}(u, n-3) = 0$;
(3) $u = v_{n-1}\bar{v}_{n-2}v_{n-3}\cdots v_0$ for $\dddot{\oplus}(u, n-3) = 1$.

Note that Definition 1 can only be applied for odd integer n. Fig. 1 depicts twisted cubes TQ_3 and TQ_5, respectively. Recently, Wang et al. [19] showed that Definition 1 can be further extended to any integer $n \geqslant 1$ by considering two types of TQ_n for even integer n as follows:

Definition 2 (*[19]*)**.** For an even integer $n \geqslant 2$, the n-dimensional twisted cube TQ_n is divided into two types: 0-type TQ_n and 1-type TQ_n, where the former is denoted by TQ^0_n and the latter is denoted by TQ^1_n. For $b \in \mathbb{Z}_2$, $V(TQ^b_n) = \{ibx\colon\ i \in \mathbb{Z}_2 \text{ and } x \in V(TQ_{n-1})\}$ and $E(TQ^b_n) = \cup_{i\in\mathbb{Z}_2}\{(ibx, iby)\colon\ (x, y) \in E(TQ_{n-1})\} \cup E'$, where an edge $(u, v) \in E'$ if and only if the two nodes $u = u_n b u_{n-2}\cdots u_0$ and $v = v_n b v_{n-2}\cdots v_0$ satisfy $u = \bar{v}_n b v_{n-2}\cdots v_0$.

Fig. 2 illustrates the two types of twisted cubes TQ^0_4 and TQ^1_4, respectively. In the rest of this paper, we say TQ_n to mean either 0-type TQ_n or 1-type TQ_n if n is even and there is no ambiguity. Also, for notational convenience, a node $x \in V(TQ_n)$ is denoted by $x = (x_n)x_{n-1}x_{n-2}\cdots x_0$, where the first bit x_n enclosed by a pair of round brackets indicates that we can omit it if n is odd. According to Definitions 1 and 2, twisted cubes can be equivalently defined by the following non-recursive fashion.

Theorem 1. *Let $n \geqslant 1$ and $v \in V(TQ_n)$. For $i \in \mathbb{Z}_n$, the ith dimensional adjacent node (or the i-neighbor) of v in TQ_n, denoted by $N_i(v)$ is computed as follows:*