Applied Mathematics and Computation 268 (2015) 489–495

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

A Fast Parallel Algorithm for Constructing Independent Spanning Trees on Parity Cubes☆

Yu-Huei Chang[a], Jinn-Shyong Yang[b], Jou-Ming Chang[c,*], Yue-Li Wang[a]

[a] Department of Information Management, National Taiwan University of Science and Technology, Taipei, Taiwan, ROC
[b] Department of Information Management, National Taipei University of Business, Taipei, Taiwan, ROC
[c] Institute of Information and Decision Sciences, National Taipei University of Business, Taipei, Taiwan, ROC

ARTICLE INFO

Keywords:
Parallel algorithms
Independent spanning trees
Parity cubes
Interconnection networks

ABSTRACT

Zehavi and Itai (1989) proposed the following conjecture: every k-connected graph has k independent spanning trees (ISTs for short) rooted at an arbitrary node. An n-dimensional parity cube, denoted by PQ_n, is a variation of hypercubes with connectivity n and has many features superior to those of hypercubes. Recently, Wang et al. (2012) confirmed the ISTs conjecture by providing an $\mathcal{O}(N \log N)$ algorithm to construct n ISTs rooted at an arbitrary node on PQ_n, where $N = 2^n$ is the number of nodes in PQ_n. However, this algorithm is executed in a recursive fashion and thus is hard to be parallelized. In this paper, we present a non-recursive and fully parallelized approach to construct n ISTs rooted at an arbitrary node of PQ_n in $\mathcal{O}(\log N)$ time using N processors. In particular, the constructing rule of spanning trees is simple and the proof of independency is easier than ever before.

1. Introduction

A set of spanning trees in a graph G is called *independent spanning trees* (ISTs for short) if all the trees are rooted at the same node r such that, for any node $v(\neq r)$ in G, 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 the two end nodes v and r. Constructing multiple ISTs in networks have been studied from not only the theoretical point of view but also some practical applications, such as fault-tolerant broadcasting [1,10] and secure message distribution [1,12,16].

For a graph G, its node set and edge set are denoted by $V(G)$ and $E(G)$, respectively. If F is a subset of $V(G)$, we denote by $G - F$ the graph obtained from G by removing F. A graph G is *k-connected* if $|V(G)| > k$ and $G - F$ is connected for every subset $F \subseteq V(G)$ with $|F| < k$. Zehavi and Itai [20] proposed the following conjecture: If G is a k-connected graph and $r \in V(G)$ is an arbitrary node, then G has k ISTs rooted at r. From then on, the conjecture has been confirmed only for k-connected graphs with $k \leq 4$ (see [6,7,10,20]), and it is still open for arbitrary k-connected graphs when $k \geq 5$. Moreover, by providing construction schemes of ISTs, the conjecture has been proved to be affirmative for several restricted classes of graphs (see recent papers [2–5,15–18] and the references quoted therein).

Wang and Zhao [13] first introduced the family of parity cubes (a synonym called twisted-cubes in that paper) as a variant of hypercubes. Note that although the synonym "twisted-cube" has a similar name to the twisted cube defined in [9], the two

☆ This research was partially supported by National Science Council under the grants MOST103-2221-E-141-001 and MOST103-2221-E-141-002.
* Corresponding author. Tel.: +886223226505.
E-mail address: spadechang@icloud.com, spade@ntub.edu.tw (J.-M. Chang).

http://dx.doi.org/10.1016/j.amc.2015.06.081
0096-3003/

graphs are completely different. In order to distinguish twisted-cubes from twisted cubes, Wang et al. [15] suggested to use parity cubes instead of twisted-cubes. The n-dimensional parity cube, denoted as PQ_n, has many features superior to those of normal hypercube Q_n. For example, Wang and Zhao [13] showed that the diameter of PQ_n is $\lceil (n+1)/2 \rceil$, which is about half that of Q_n for a sufficiently large dimension n. So far the study of properties for PQ_n has received less attention except for [11,14,15]. Park et al. [11] proved that PQ_n is $(n-2)$-hamiltonian and $(n-3)$-hamiltonian connected for any integer $n \geq 3$. Wang et al. [14] provided embedding schemes such that meshes can be embedded into PQ_n. Since it has been proved in [13] that PQ_n has the connectivity n, Wang et al. [15] proposed an algorithm to construct n ISTs rooted at an arbitrary node in $\mathcal{O}(N \log N)$ time for PQ_n, where $N = 2^n$ is the number of nodes in PQ_n. However, this algorithm is executed in a recursive fashion and thus is hard to be parallelized. In this paper, we present a non-recursive and fully parallelized approach for constructing n ISTs rooted at an arbitrary node in PQ_n.

The rest of this paper is organized as follows. Section 2 formally gives the definition of parity cubes and provides some useful terminologies and notations. Section 3 presents our algorithm to construct n ISTs in PQ_n. Section 4 proves the correctness of the algorithm. The final section contains our concluding remarks.

2. Preliminaries

Let $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$. For any integer $n \geq 1$, a binary string x of length n will be written as $x_{n-1}x_{n-2}\cdots x_0$, where $x_i \in \{0, 1\}$ for each $i \in \mathbb{Z}_n$. In particular, x_i is called the ith bit of x and $x_{n-1}x_{n-2}\cdots x_i$ for each $i \in \mathbb{Z}_n$ is called a *prefix* of x. The complement of x_i is denoted by $\bar{x}_i$. For two binary strings x and y of length n, we define $x + y = z_{n-1}z_{n-2}\cdots z_0$, where $z_i = (x_i + y_i) \pmod 2$ for any $i \in \mathbb{Z}_n$. That is, the operation "+" in $x + y$ is the modulo 2 addition. Also, we use e_i to denote an n-bit binary string whose ith bit is 1 and all the other bits are 0. Clearly, $x + e_i = x_{n-1}x_{n-2}\cdots x_{i+1}\bar{x}_i x_{i-1}\cdots x_0$. Furthermore, for any integer $p \in \{0, 1\}$, we define $p \cdot x = x'_{n-1}x'_{n-2}\cdots x'_0$ where $x'_i = p \cdot x_i \pmod 2$.

Similar to the n-dimensional hypercube, the n-dimensional parity cube PQ_n is an n-regular graph of 2^n nodes in which each node is labeled by a unique binary string of length n. Originally, PQ_n can be recursively defined as follow.

Definition 1. [13] The n-dimensional parity cube, PQ_n, is recursively defined as follows:

(1) PQ_1 is defined as the complete graph with two vertices labeled 0 and 1. PQ_2 is a graph consisting of four vertices labeled with 00, 01, 10, and 11, respectively, and connected by four edges (00, 01), (00, 10), (01, 11), and (10, 11). PQ_3 is the twisted 3-cube.

(2) For $n \geq 4$, PQ_n is constructed from two disjoint copies of PQ_{n-1}, according to the following rules. For $i \in \{0, 1\}$, let PQ^i_{n-1} denote the graph obtained from PQ_{n-1} by adding a prefix bit i in the front of the label of each node $x \in V(PQ_{n-1})$. Then connect each node $u = u_{n-1}u_{n-2}\cdots u_0$ in PQ^0_{n-1} with the node $v = v_{n-1}v_{n-2}\cdots v_0$ in PQ^1_{n-1} as follows:

$$v = u + e_{n-1} + \sum_{k=0}^{\lfloor \frac{n-3}{2} \rfloor} u_{2k} \cdot e_{2k+1} + \sum_{k=2}^{n-2} u_k \cdot e_1.$$

Fig. 1 demonstrates PQ_3 and PQ_4, respectively.

In what follows, a substitutional definition of parity cubes will be given. Two binary strings $x = x_1x_0$ and $y = y_1y_0$ are *pair-related*, denoted by $x \sim y$, if and only if $(x, y) \in \{(00, 00), (10, 10), (01, 11), (11, 01)\}$. Note that this definition has been used to define another variant of hypercubes called crossed cubes [8]. According to the definition of pair relation, parity cubes can be equivalently defined by the following non-recursive fashion:

Definition 2. Let $n \geq 1$ be an integer. Two nodes $x = x_{n-1}x_{n-2}\cdots x_0$ and $y = y_{n-1}y_{n-2}\cdots y_0$ are joined by an edge in PQ_n if and only if there exists an integer $i \in \mathbb{Z}_n$ such that

(1) $y_{n-1}y_{n-2}\cdots y_{i+1} = x_{n-1}x_{n-2}\cdots x_{i+1}$;
(2) $y_i \neq x_i$;
(3) $y_{i-1} = x_{i-1}$ if i is odd;
(4) If $i \geq 2$, then
  (4.1) $y_{2j+1}y_{2j} \sim x_{2j+1}x_{2j}$ for $0 < j < \lfloor i/2 \rfloor$;
  (4.2) $y_1 = \begin{cases} \sum_{j=0}^{i-1} x_j, & \text{if } x_i = 0; \\ x_{i-1} + x_0 + \sum_{j=1}^{\lfloor \frac{i-1}{2} \rfloor} x_{2j-1}, & \text{otherwise;} \end{cases}$
  (4.3) $y_0 = x_0$.

In the above definition, y is said to be the *i-neighbor* of x, and for notational convenience we write $y = N_i(x)$. For example, if we consider the node $x = 011011_2 = 27$ in PQ_6, then $N_0(x) = 011010$, $N_1(x) = 011001$, $N_2(x) = 011101$, $N_3(x) = 010001$, $N_4(x) = 001011$ and $N_5(x) = 111001$.

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