Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

Constructing completely independent spanning trees in crossed cubes

Baolei Cheng [a,b], Dajin Wang [c], Jianxi Fan [a,b,*]

[a] *School of Computer Science and Technology, Soochow University, Suzhou 215006, China*
[b] *Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou 215006, China*
[c] *Department of Computer Science, Montclair State University, Upper Montclair, NJ 07043, USA*

ARTICLE INFO

Article history:
Received 14 April 2016
Received in revised form 16 November 2016
Accepted 16 November 2016
Available online xxxx

Keywords:
Completely independent spanning tree
Crossed cube
Fault tolerance
Reliable broadcasting

ABSTRACT

The *Completely Independent Spanning Trees* (*CISTs*) are a very useful construct in a computer network. It can find applications in many important network functions, especially in reliable broadcasting, i.e., guaranteeing broadcasting operation in the presence of faulty nodes. The question for the existence of two CISTs in an arbitrary network is an NP-hard problem. Therefore most research on CISTs to date has been concerning networks of specific structures. In this paper, we propose an algorithm to construct two CISTs in the *crossed cube*, a prominent, widely studied variant of the well-known hypercube. The construction algorithm will be presented, and its correctness proved. Based on that, the existence of two CISTs in a special Bijective Connection network based on crossed cube is also discussed.

1. Introduction

The *Completely Independent Spanning Trees* (CISTs for short) of a computer network is a set of spanning trees $\{T_1, T_2, \ldots, T_k\}$ of the network, such that for any two nodes u and v, the u-to-v paths belonging to T_i and T_j respectively have no common intermediate nodes, for all $i \neq j$. CISTs can find applications in important network functions such as multi-node broadcasting [3], one-to-all broadcasting [33], reliable broadcasting, secure message distribution [14,20], and increase of bandwidth [19]. For example, the network bandwidth can be increased by using the multiple paths between any two nodes in CISTs. Some examples of CISTs are illustrated in Fig. 1. The pairs in (a) and (b) are CISTs, while the pair in (c) is not because the paths from u to v in both trees contain node w.

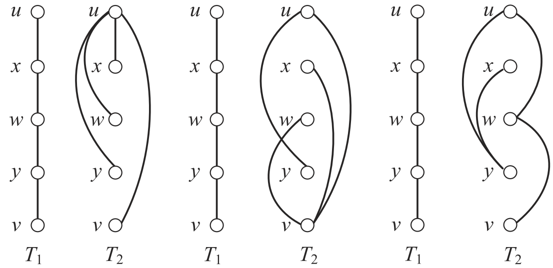
The decision problem as to whether there exist two CISTs in a general graph G has been shown NP-hard [15], and conditions have been proposed for CIST-containing graphs of certain structures. It was once conjectured that a k-connected graph ($k \geq 2$) always contains two CISTs, only to be disproved recently by Péterfalvi and Pai [24,31]. Therefore most research on CISTs to date has been concerning networks of specific structures. Especially, the problem of constructing multiple CISTs in a certain regularly-structured network has received much attention [15,24,7,16,18,28].

As an important variant of the hypercube, the best-known structure for the interconnection network, the *crossed cube* (CQ_n for short) has attracted the interest of many researchers [6,5,4,8,12,25,26,29,30,34,35]. It was proved in [21] that an n-dimensional crossed cube CQ_n is not node-transitive, and [22] showed that the node set of CQ_n can be divided into $2^{\lceil (n-4)/2 \rceil}$ equivalence classes, where the nodes in one equivalence class are all similar. Since there are two equivalence classes in the

* Corresponding author at: School of Computer Science and Technology, Soochow University, Suzhou 215006, China.
E-mail addresses: chengbaolei@suda.edu.cn (B. Cheng), wangd@mail.montclair.edu (D. Wang), jxfan@suda.edu.cn (J. Fan).

http://dx.doi.org/10.1016/j.dam.2016.11.019
0166-218X/

Please cite this article in press as: B. Cheng, et al., Constructing completely independent spanning trees in crossed cubes, Discrete Applied Mathematics (2016), http://dx.doi.org/10.1016/j.dam.2016.11.019



(a) T_1, T_2 are CISTs. (b) T_1, T_2 are CISTs. (c) T_1, T_2 are not CISTs.

Fig. 1. Example of CISTs in (a) and (b); non-CISTs in (c).

n-dimensional locally twisted cube LTQ_n [27], and there are only one equivalence class [32] in the n-dimensional hypercube Q_n, CQ_n is considered more complex than LTQ_n and Q_n.

Some characterizations have been proposed for the existence of CISTs in arbitrary graphs. Recently, Araki [1] proved that there exist two CISTs in a graph with $n \geq 3$ nodes if the degree of each node is $\geq n/2$, and Fan et al. [11] proved the existence of two CISTs in a graph with $n \geq 3$ nodes if the sum of degrees of any two nodes is $\geq n$. H.-Y. Chang et al. proved that for graphs of order n, with $n \geq 6$, if the minimum degree is at least $n - 2$, then there are at least $\lfloor n/3 \rfloor$ CISTs [2]. T. Hasunuma showed that for any graph G with $n \geq 7$ nodes, if the minimum degree of a vertex is $n-k$, where $3 \leq k \leq n/2$, then there are $\lfloor n/k \rfloor$ CISTs in G [17]. These are rather strong sufficient conditions because the needed degree of every node/pair of nodes is very large comparing the number of nodes. Furthermore, they are all existential proofs rather than constructional.

In this paper, we study the problem of constructing CISTs in a CQ_n, the n-dimensional crossed cube. More specifically, we will (a) show that there exist two CISTs in a CQ_n if $n \geq 4$; (b) present a recursive algorithm to construct two CISTs in a CQ_n, $n \geq 4$, with the diameter of either tree being no more than $n + 4$. Furthermore, we will discuss the existence of CISTs in a more generalized network – the CQ-based BC network – and show that it contains two CISTs.

The rest of this paper is organized as follows. In Section 2, we introduce terminology and notation that will be used throughout the paper, define the crossed cube CQ_n, and give some relevant properties of the CQ_n. Section 3 presents the main work of the paper: Proposing the algorithm that constructs two CISTs in a CQ_n, and proving the correctness of the algorithm. Section 4 discusses the existence of CISTs in CQ-based BC networks. Section 5 gives summarizing remarks and concludes the paper.

2. Preliminaries

2.1. Graph terminology and notation

A computer interconnection network can be represented by a graph, where nodes represent processors and edges represent links between processors. Given a graph G, we use $V(G)$ and $E(G)$ to denote the node set and the edge set, respectively, in G. Let S be a nonempty subset of $V(G)$. The subgraph of G *induced by* S, denoted by $G[S]$, is the subgraph with the node set S and those edges of G with both ends in S. A graph G_1 is said to be *isomorphic* to graph G_2, if there is a bijection ψ: $V(G_1) \rightarrow V(G_2)$ such that $(\psi(x), \psi(y)) \in E(G_2)$ if and only if $(x, y) \in E(G_1)$. Let $u, v \in V(G)$. A path from u to v is called $\langle u, v\rangle$-path, denoted by R: $u = x^{(0)}, x^{(1)}, \ldots, x^{(k)} = v$, where $(x^{(i)}, x^{(i+1)}) \in E(G))$ for all $0 \leq i \leq k - 1$. R can also be denoted by R: $u = x^{(0)}, x^{(1)}, \ldots, x^{(i-1)}, R', x^{(j+1)}, x^{(j+2)}, \ldots, x^{(k)} = v$, where R' is a $\langle x^{(i)}, x^{(j)}\rangle$-path, a *subpath* of R. The subpath R', starting from $x^{(i)}$ and ending with $x^{(j)}$, can be denoted by $R' = (R; x^{(i)}, x^{(j)})$. We use $V(R)$ and $E(R)$ to denote the node set and the edge set in R, respectively. Two $\langle x, y\rangle$-paths P and Q are *edge-disjoint* if $E(P) \cap E(Q) = \emptyset$. Two $\langle x, y\rangle$-paths P and Q are *internally node-disjoint* if they are edge-disjoint and $V(P) \cap V(Q) = \{x, y\}$. Two spanning trees T_1 and T_2 rooted at a node u in graph G are *independent* if the $\langle u, v\rangle$-path in T_1 and the $\langle u, v\rangle$-path in T_2 are internally node-disjoint for each $v \in V(G) \setminus \{u\}$. If for any two nodes x and y in $V(G)$, the $\langle x, y\rangle$-path in T_1 and the $\langle x, y\rangle$-path in T_2 are internally node-disjoint, then T_1 and T_2 are said to be *completely independent*. A set of spanning trees in G are independent (completely independent) if they are pairwise independent (completely independent). In this paper, we will use CISTs (ISTs) to refer to completely independent spanning trees (independent spanning trees).

2.2. The crossed cube

Let $b_{n-1}b_{n-2}\ldots b_{i+1}b_ib_{i-1}\ldots b_0$ be a binary string of length n, where $b_i \in \{0, 1\}$. b_i is called the ith bit of the string, with b_{n-1} being the most significant bit and b_0 the least significant bit. An n-dimensional crossed cube denoted CQ_n, has 2^n nodes. Each node in CQ_n is represented by a unique binary string of length n, called the *address* of the node. The node addresses range from $\underbrace{00\ldots0}_{n}$ to $\underbrace{11\ldots1}_{n}$.