Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2014, Article ID 498349, 4 pages
http://dx.doi.org/10.1155/2014/498349

Research Article

Edge-Disjoint Fibonacci Trees in Hypercube

Indhumathi Raman[1] and Lakshmanan Kuppusamy[2]

[1] *School of Information Technology & Engineering, VIT University, Vellore 632 014, India*
[2] *School of Computing Science & Engineering, VIT University, Vellore 632 014, India*

Correspondence should be addressed to Indhumathi Raman; indhumathi.raman@gmail.com

Received 30 June 2014; Accepted 7 October 2014; Published 19 October 2014

Academic Editor: Liansheng Tan

The Fibonacci tree is a rooted binary tree whose number of vertices admit a recursive definition similar to the Fibonacci numbers. In this paper, we prove that a hypercube of dimension h admits two edge-disjoint Fibonacci trees of height h, two edge-disjoint Fibonacci trees of height $h-2$, two edge-disjoint Fibonacci trees of height $h-4$ and so on, as subgraphs. The result shows that an algorithm with Fibonacci trees as underlying data structure can be implemented concurrently on a hypercube network with no communication latency.

1. Introduction

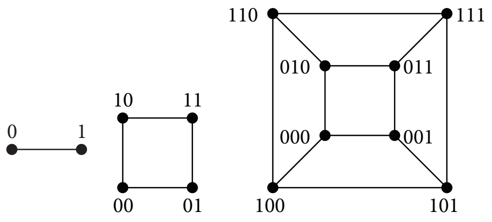
High speed computing is a key requirement for many applications of science and technology. Parallel computers perform high speed computing with thousands of processors operating parallelly and concurrently. The interconnection network of a parallel computer connects various components of the system and plays a crucial role in the computational process. A natural way to represent this interconnection network is to construct a graph where each processor of the system is represented by a vertex and each link connecting two processors is represented by an edge. This representation is the genesis for the interdisciplinary research involving engineering and computational aspects of parallel computers and graph theory.

Among the interconnection networks of parallel computers, the binary *hypercube* has received much attention. An important property of a hypercube, which makes it popular, is its ability to efficiently simulate the message routings of other interconnection networks. Defining mathematically, an *n-dimension hypercube*, Q_n, has 2^n vertices each labelled with a binary string of length n. Two vertices are adjacent if and only if their labels differ in exactly one position. Figure 1 shows hypercubes of dimensions 1, 2, and 3.

The problem of efficiently implementing parallel algorithms on parallel computers has been studied as a graph embedding problem, which is to embed the communication graph G (underlying a parallel algorithm A) within the interconnection network topology H (underlying a parallel computer N) with minimum communication overhead. Graph embeddings are mathematical models capturing the issues involved in the implementation of parallel algorithms on a parallel computer (see [1]) and establishing equivalence between interconnection networks of two parallel computers (see [2, 3]).

Formally defining, an *embedding* ϕ of (guest) graph $G(V,E)$ into (host) graph $H(W,F)$ is a function $\phi : V \rightarrow W$ such that if (u,v) is an edge in G, then $\phi(u)$ and $\phi(v)$ are connected by a path in H. Dilation is one of the important parameters that measure the quality of an embedding. Defining mathematically, dilation $(\phi) := \max\{\mathrm{dist}_H(\phi(u),\phi(v)) : (u,v) \in E\}$, where $\mathrm{dist}_H(x,y)$ denotes the length of the shortest path connecting two vertices x and y in W in H. The dilation measures the communication delay when graph G is embedded on graph H. If dilation equals 1, then it implies that there is no communication latency and G is isomorphic to a subgraph of H and we write $G \subseteq H$. Since one aims to reduce the communication latency in a network, the goal of a parallel algorithm designer is to map the algorithm graph G into the corresponding network H such that $G \subseteq H$.

From a computing perspective, trees form an important class of computational structures. They naturally arise in



(a) Q_1 (b) Q_2 (c) Q_3

Figure 1: Hypercubes of small dimension.

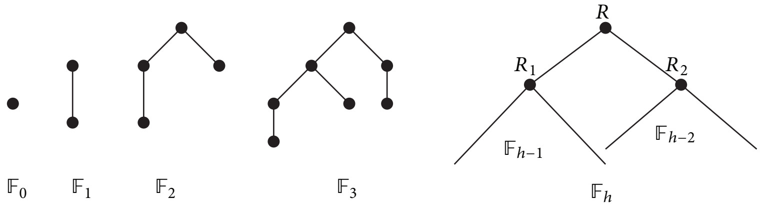


Figure 2: Examples of Fibonacci trees and recursive construction.

the design of parallel algorithms which require basic operations like merging, sorting, and searching. Hence, there is a large literature on embedding of various kinds of trees into the graphs of interconnection networks. In particular, embedding of binary trees into hypercubes has received special attention since trees serve as computational structures for parallel algorithms that employ, divide, and conquer paradigm.

Several researchers work on the problem of embedding a set of subgraphs, such as paths, rings, and trees, satisfying some constraints in a network. The problem of finding edge-disjoint spanning trees is an important example [4–6]. A number of trees of a graph H are said to be edge-disjoint if no two trees contain the same edge of H. Besides being of theoretical interest, this problem has two practical applications. One is to enhance the ability of fault tolerance. One way to achieve fault tolerant interprocessor communication is by exploiting and effectively utilizing the disjoint paths that exist between pairs of source and destination. The other application is to develop efficient collective communication algorithms in distributed memory parallel computers [5]. Several researches have been done to recognise what trees and how many of them can be accommodated in the hypercube; see [4–6].

It has been proved in [7] that a Fibonacci tree $\mathbb{F}_h$ of height h is a subgraph of a hypercube with dimension $\lfloor 0.75h \rfloor + 1.5$. Since the hypercube $Q_{\lfloor 0.75h \rfloor + 1.5}$ is not big enough to accommodate two copies of $\mathbb{F}_h$, we choose Q_h to embed at least two copies of edge-disjoint Fibonacci trees. In this paper, we prove that, for $h \geq 3$, the hypercube Q_h contains edge-disjoint copies of two $\mathbb{F}_h$, two $\mathbb{F}_{h-2}, \ldots$, and two $\mathbb{F}_i$, where $i = 2$ if h is even and $i = 3$ if h is odd. This is denoted by $Q_h \supseteq 2\mathbb{F}_h \cup 2\mathbb{F}_{h-2} \cup \cdots \cup 2\mathbb{F}_i$. For any graph G, an embedding $L : G \rightarrow Q_n$ is a labelling of the vertices of G with binary strings of length n. Clearly, $G \subseteq Q_n$ if and only if $L(u)$ and $L(v)$ differ in exactly one position whenever (u, v) is an edge in G. This labelling technique is widely used in our proofs of embedding.

2. Edge-Disjoint Embedding of Fibonacci Trees

A formal recursive definition of the Fibonacci tree (denoted by $\mathbb{F}_h$ if its height is h) is given below.

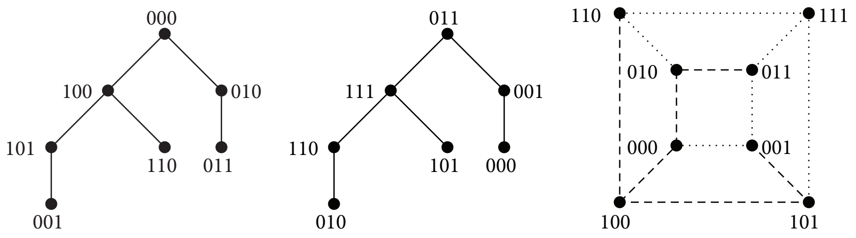
Definition 1. $\mathbb{F}_0 := K_1$, $\mathbb{F}_1 := K_2$, where K_i is a complete graph on i vertices. For $h \geq 2$, $\mathbb{F}_h$ is obtained by taking a copy of $\mathbb{F}_{h-1}$, a copy of $\mathbb{F}_{h-2}$, and a new vertex R and joining R to the roots of $\mathbb{F}_{h-1}$ and $\mathbb{F}_{h-2}$.

Figure 2 shows a few examples of small Fibonacci trees and the recursive construction.

The above recursive definition implies that the number of vertices in $\mathbb{F}_h$ is $|V(\mathbb{F}_h)| = |V(\mathbb{F}_{h-1})| + |V(\mathbb{F}_{h-2})| + 1$. On solving this recurrence relation, we get $|V(\mathbb{F}_h)| = f(h+2) - 1$, where $f(i)$ is the ith number in the Fibonacci sequence, $f(0) = 1$, $f(1) = 1$, $f(n) = f(n-1) + f(n-2)$; this justifies the terminology Fibonacci tree. The number of vertices of the Fibonacci tree has helped to give a closed-form representation of the Fibonacci numbers; see [8].

Theorem 2. *For every $h \geq 3$, the hypercube Q_h contains two edge-disjoint Fibonacci trees $\mathbb{F}_h^L$ and $\mathbb{F}_h^R$ of height h such that roots of $\mathbb{F}_h^L$ and $\mathbb{F}_h^R$ are labelled 0^h and $0^{h-2}11$, respectively.*

Proof. We prove the result by induction on h. To do so, it is sufficient to give two labellings $B_h^L : \mathbb{F}_h^L \rightarrow Q_h$ and $B_h^R : \mathbb{F}_h^R \rightarrow Q_h$ for the tree $\mathbb{F}_h$. For the basic case $h = 3, 4$, the two labellings for each of $\mathbb{F}_3^L$ and $\mathbb{F}_4^L$ are given in Figures 3 and 4, respectively. In Figure 3(c) (resp., Figure 4(c)), we show the two trees $\mathbb{F}_3^L$ and $\mathbb{F}_3^R$ (resp., $\mathbb{F}_4^L$ and $\mathbb{F}_4^R$) inside Q_3 (resp., Q_4) as dashed and dotted lines, respectively.

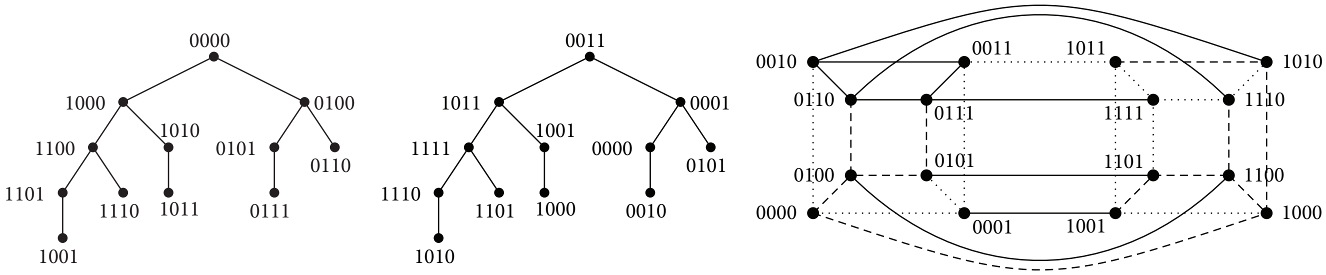


(a) $B_3^L : \mathbb{F}_3^L \rightarrow Q_3$

(b) $B_3^R : \mathbb{F}_3^R \rightarrow Q_3$

(c) $\mathbb{F}_3^L$ and $\mathbb{F}_3^R$ in Q_3 are shown in dashed and dotted lines, respectively

Figure 3: Embedding $\mathbb{F}_3^L$ and $\mathbb{F}_3^R$ in Q_3.

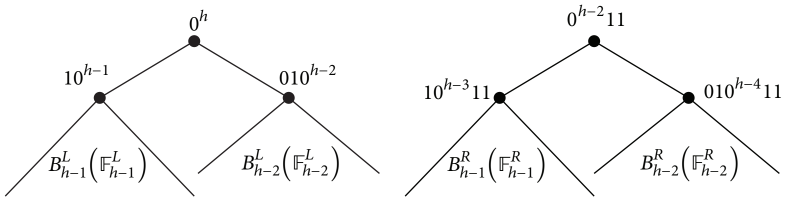


(a) $B_4^L : \mathbb{F}_4^L \rightarrow Q_4$

(b) $B_4^R : \mathbb{F}_4^R \rightarrow Q_4$

(c) $\mathbb{F}_4^L$ and $\mathbb{F}_4^R$ in Q_4 are shown in dashed and dotted lines, respectively

Figure 4: Embedding $\mathbb{F}_4^L$ and $\mathbb{F}_4^R$ in Q_4.


0^h
10^{h-1}
010^{h-2}
$B_{h-1}^L(\mathbb{F}_{h-1}^L)$
$B_{h-2}^L(\mathbb{F}_{h-2}^L)$
$0^{h-2}11$
$10^{h-3}11$
$010^{h-4}11$
$B_{h-1}^R(\mathbb{F}_{h-1}^R)$
$B_{h-2}^R(\mathbb{F}_{h-2}^R)$

(a) Labelling $B_h^L : \mathbb{F}_h^L \rightarrow Q_h$

(b) Labelling $B_h^R : \mathbb{F}_h^R \rightarrow Q_h$

Figure 5: Two labellings B_h^L and B_h^R for $\mathbb{F}_h$.

For $h \geq 5$, we give a labelling procedure for $\mathbb{F}_h^L$ and $\mathbb{F}_h^R$. We first provide the labelling for $\mathbb{F}_h^L$. By induction hypothesis we are given embeddings (labellings) $B_{h-1}^L : \mathbb{F}_{h-1}^L \rightarrow Q_{h-1}$ and $B_{h-2}^L : \mathbb{F}_{h-2}^L \rightarrow Q_{h-2}$. For notational convenience, we denote $B_{h-1}^L(\mathbb{F}_{h-1}^L)$ by B_{h-1}^L and $B_{h-2}^L(\mathbb{F}_{h-2}^L)$ by B_{h-2}^L. We extend the labellings B_{h-1}^L and B_{h-2}^L to a labelling B_h^L of $\mathbb{F}_h^L$ by prefixing the labels of all the vertices of $\mathbb{F}_{h-1}^L$ by 1 and the labels of all the vertices of $\mathbb{F}_{h-2}^L$ by 01 and label the root of $\mathbb{F}_h^L$ by 0^h; see Figure 5(a). We note that the roots of $\mathbb{F}_{h-1}^L$ and $\mathbb{F}_{h-2}^L$ are labelled 10^{h-1} and 010^{h-2} and both are adjacent to the root of $\mathbb{F}_h^L$ which is labelled 0^h. In a similar way, we next provide the labelling for $\mathbb{F}_h^R$. By induction hypothesis we are given embeddings (labellings) $B_{h-1}^R : \mathbb{F}_{h-1}^R \rightarrow Q_{h-1}$ and $B_{h-2}^R : \mathbb{F}_{h-2}^R \rightarrow Q_{h-2}$. For notational convenience, we denote $B_{h-1}^R(\mathbb{F}_{h-1}^R)$ by B_{h-1}^R and $B_{h-2}^R(\mathbb{F}_{h-2}^R)$ by B_{h-2}^R. We extend the labellings B_{h-1}^R and B_{h-2}^R to a labelling B_h^R of $\mathbb{F}_h^R$ by prefixing the labels of all the vertices of $\mathbb{F}_{h-1}^R$ by 1 and the labels of all the vertices of $\mathbb{F}_{h-2}^R$ by 01 and label the root of $\mathbb{F}_h^R$ by $0^{h-1}11$; see Figure 5(b). The recursive labelling technique of $\mathbb{F}_h^L$ (and hence $\mathbb{F}_h^R$) does not alter the adjacency property of the vertices since B_h^L (and hence B_h^R) is an injective extension of B_{h-1}^L and B_{h-2}^L (B_{h-1}^R and B_{h-2}^R). The trees $\mathbb{F}_h^L$ and $\mathbb{F}_h^R$ are the required two edge-disjoint copies of F_h^* in Q_h. □

Theorem 3. *For $h \geq 3$, $Q_h \supseteq 2\mathbb{F}_h \cup 2\mathbb{F}_{h-2} \cup \cdots \cup 2\mathbb{F}_i$, where $i = 2$ if h is even and $i = 3$ if h is odd.*

Proof. In the labelling technique of $\mathbb{F}_h$ discussed in Theorem 2, the root is the only vertex whose label starts with 00. Hence, apart from the two copies of $\mathbb{F}_h$, the edges of $00Q_{h-2}$ are unused in Q_h. By recursion, Q_{h-2} contains $2\mathbb{F}_{h-2} \cup 2\mathbb{F}_{h-4} \cup \cdots \cup 2\mathbb{F}_i$, hence the theorem. As an illustration, referring to Figure 4, the hypercube Q_4 apart from containing two $\mathbb{F}_4$ shown in dashed and dotted lines, it also contains two $\mathbb{F}_2$ as follows: since $\mathbb{F}_2$ is isomorphic to a path on four vertices, we give two path labellings (1) 1110 − 0110 − 0010 − 1010 and (2) 0010 − 0011 − 0111 − 1111. □

Remark 4. An induced subgraph of the hypercube based on the Fibonacci numbers is called the *Fibonacci Cube*. This interconnection network was introduced by Hsu in [9]. Several topological properties of the Fibonacci cube have been studied over the recent years and various extensions have been introduced; see [10, 11]. Also, recently there has been some work done on graph indexing and we refer to [12–14].

3. Conclusion

In this paper we proved that, for every $h \geq 3$, the hypercube Q_h of dimension h contains two edge-disjoint copies of $\mathbb{F}_h, \mathbb{F}_{h-2}, \ldots \mathbb{F}_2$ as subgraph (that is with dilation 1). This result shows that an algorithm with Fibonacci tree as its underlying data structure can be made to run concurrently on a hypercube network with no communication latency. Thus, this mathematical result will have a significant interest in the fields of parallel computing and interconnection networks.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] S. N. Bhatt and I. C. F. Ipsen, "How to embed trees in hypercubes," Research Report YALEU/DCS/RR-443, Yale University, Department of Computing Science, 1985.

[2] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan Kaufmann, San Mateo, Calif, USA, 1992.

[3] B. Monien and H. Sudborough, "Embedding one interconnection network in another," *Computing Supplementary*, vol. 7, pp. 257–282, 1990.

[4] S. Wagner and M. Wild, "Decomposing the hypercube Q_n into n isomorphic edge-disjoint trees," *Discrete Mathematics*, vol. 312, no. 10, pp. 1819–1822, 2012.

[5] B. Barden, R. Libeskind-Hadas, J. Davis, and W. Williams, "On edge-disjoint spanning trees in hypercubes," *Information Processing Letters*, vol. 70, no. 1, pp. 13–16, 1999.

[6] J. F. Fink, "On the decomposition of n-cubes into isomorphic trees," *Journal of Graph Theory*, vol. 14, no. 4, pp. 405–411, 1990.

[7] S. A. Choudum and I. Raman, "Embedding height balanced trees and Fibonacci trees in hypercubes," *Journal of Applied Mathematics and Computing*, vol. 30, no. 1-2, pp. 39–52, 2009.

[8] I. Raman, "A note on closed-form representation of Fibonacci numbers using Fibonacci trees," *ISRN Discrete Mathematics*, vol. 2014, Article ID 132925, 3 pages, 2014.

[9] W. J. Hsu, "Fibonacci Cubes. A new interconnection topology," *IEEE Transactions on Parallel and Distributed Systems*, vol. 4, no. 1, pp. 3–12, 1993.

[10] S. Klavžar, "On median nature and enumerative properties of Fibonacci-like cubes," *Discrete Mathematics*, vol. 299, no. 1–3, pp. 145–153, 2005.

[11] J. Wu, "Extended fibonacci cubes," *IEEE Transactions on Parallel and Distributed Systems*, vol. 8, no. 12, pp. 1203–1210, 1997.

[12] R. Sun, Z. Zhu, and L. Tan, "On the Merrifield-Simmons index and Hosoya index of bicyclic graphs with a given girth," *Ars Combinatoria*, vol. 103, pp. 465–478, 2012.

[13] Z. Zhu, L. Tan, and Z. Qiu, "Tricyclic graph with maximal Estrada index," *Discrete Applied Mathematics*, vol. 162, pp. 364–372, 2014.

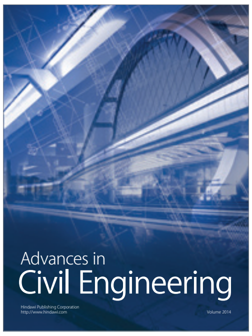
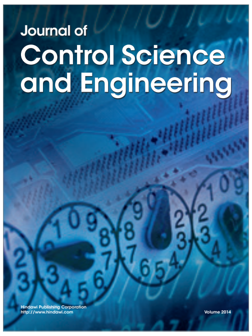
[14] Z. Zhu, T. Tao, J. Yu, and L. Tan, "On the Harary index of cacti," *FILOMAT*, vol. 28, no. 3, pp. 493–507, 2014.

Journal of
Engineering

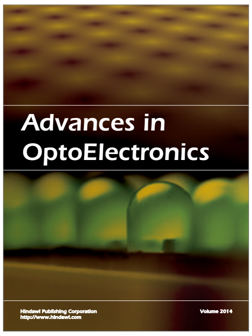
The Scientific World Journal

International Journal of
Rotating Machinery

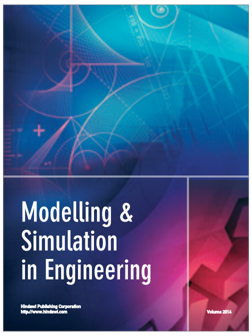
Journal of
Sensors

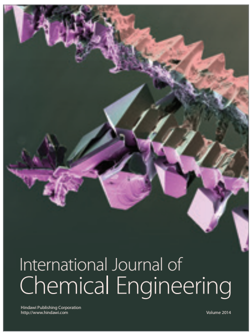
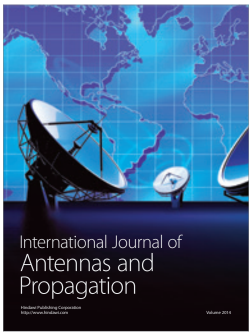
International Journal of
Distributed Sensor Networks

Advances in
Civil Engineering

Journal of
Control Science and Engineering

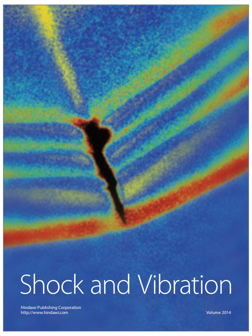
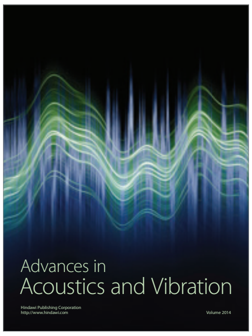
Journal of
Robotics

Journal of
Electrical and Computer Engineering

Advances in
OptoElectronics

VLSI Design

International Journal of
Navigation and Observation

Modelling & Simulation in Engineering

International Journal of
Aerospace Engineering

International Journal of
Chemical Engineering

International Journal of
Antennas and Propagation

Active and Passive Electronic Components

Shock and Vibration

Advances in
Acoustics and Vibration