How to Systematically Embed Cycles in Balanced Hypercubes

Hsuan-Han Chang, Department of Computer Science and Information Engineering, National Dong Hwa University, Shoufeng, Taiwan

Kuan-Ting Chen, Department of Computer Science and Information Engineering, National Dong Hwa University, Shoufeng, Taiwan

Pao-Lien Lai, Department of Computer Science and Information Engineering, National Dong Hwa University, Shoufeng, Taiwan

ABSTRACT

The balanced hypercube is a variant of the hypercube structure and has desirable properties like connectivity, regularity, and symmetry. The cycle is a popular interconnection topology and has been widely used in distributed-memory parallel computers. Moreover, parallel algorithms of cycles have been extensively developed and used. The problem of how to embed cycles into a host graph has attracted a great attention in recent years. However, there is no systematic method proposed to generate the desired cycles in balanced hypercubes. In this paper, the authors develop systematic linear time algorithm to construct cycles and Hamiltonian cycles for the balanced hypercube.

KEYWORDS

Balanced Hypercube, Double Reflected Edge Label Sequence, Embedding, Hamiltonian Cycle, Generalized Gray Code

1. INTRODUCTION

The balanced hypercube, a variation of the hypercube with better properties than the hypercube with the same number of vertices, was proposed by Wu and Huang (1997). The balanced hypercube is an edge transitive and vertex transitive graph. More desirable properties of balanced hypercubes were studied. This topic has been covered many times in past studies. Huang and Wu (1995) have shown that the balanced hypercube can be implemented at least as efficient as the standard hypercube in an area layout and more efficient in a linear layout. Huang and Wu (1997) have studied the resource placement problem in balanced hypercubes. Xu, Hu, and Xu (2007) have proved that the balanced hypercubes is Hamiltonian laceable. Yang (2010) has proposed that the balanced hypercube is edge-bipancyclic and Hamiltonian laceable.

A path or cycle structure is a fundamental network for multiprocessor systems and suitable for developing simple algorithms with low communication costs. Several efficient algorithms have been designed with respect to cycle structures for solving a variety of algebraic problems, graph problems, and some parallel applications, such as those in image and signal processing (Akl 1997). In the article by Baoyuan and Yizhou, the authors have studied efficient GPU-based parallel algorithms for H-tree

based data cube operations. Jianxin, Qilong and Jianer (2011) have studied the k-path problem. Leighton (1992) has proposed some parallel algorithms and Architectures.

To carry out a cycle structure algorithm on a multiprocessor computer, the processes of the parallel algorithm need to be mapped to the vertices of the interconnection network in the system so any two adjacent processes in the cycle are mapped to two adjacent vertices of the network. With regard to cycle embedding of interconnection networks, many interesting results have been obtained (Chang, Sung, & Hsu, 2000; Cheng, Hao, & Feng, 2014; Chen, Lai, & Tsai, 2010; Chen, Hsu, & Tan, 2006; Fan, Lin, & Jia, 2005; Fan, Jia, & Lin, 2008; Hao, Zhang, Feng, & Zhou, 2014; Kueng, Lin, Liang et al., 2008; Li, Peng, & Chu, 2002; Wang, 2008).

In particular, Zheng and Latifi (1997) introduced the notion of the reflected edge label sequences and proposed a kind of codeword, termed the generalized Gray code. In this paper, we extend the concepts of reflected edge label sequences and cycle patterns in Chen et al. (2010), Zheng and Latifi (1997), then propose, efficient algorithms for embedding cycles in balanced hypercubes.

The rest of this paper is organized as follows. First, we present the background and definitions relating to the balanced hypercube in Section 2. In Section 3, we propose alternated edge label sequence, reflected edge label sequence, double reflected edge label sequence, complete reflected edge label sequence, and independent alternated permutation, then demonstrate how to embed cycles employing each in the balanced hypercube. Finally, we propose zero alternated permutation and study how to construct Hamiltonian cycles using it in Section 4. The conclusion is given in the final section.

2. PRELIMINARIES

In this section we give the basic graph-theory definitions relevant to this paper. The topology of an interconnection network is conveniently represented by an undirected simple graph $G = \left(V, \ E \right)$, where $V = V \left(G \right)$ and $E = E \left(G \right)$ are the vertex set and the edge set of G, respectively. Throughout this paper, the terms graph and network are used interchangeably. Moreover, the terms node and vertex are used interchangeably. For graph terminology and notation not defined here we refer the reader to Leighton (1992) and Xu (2001).

A *walk* in a graph is a finite sequence ω: $\left(\lambda_0, e_1, \lambda_1, e_2, \ldots, \lambda_{l-1}, e_l, \lambda_l \right)$ whose terms are alternately vertices and edges so, for $1 \le i \le l$, the edge e_i has ends λ_{i-1} and λ_i, thus each edge e_i is immediately preceded and succeeded by the two vertices with which it is incident. In particular, a walk ω is called a *path* if all internal vertices, λ_i for $1 \le i \le l-1$, of the walk ω are distinct. The first vertex λ_0 of ω is called its start vertex, and the vertex λ_l is termed the last vertex. Both are called end-vertices of the path ω. For simplicity, the path ω is also denoted by $\left(\lambda_0, \lambda_1, \ldots, \lambda_i \right)$. If $\lambda_0 = \lambda_l$, ω is called a *cycle*. A cycle traversing each vertex of G exactly once is a Hamiltonian cycle.

Wu and Huang (1997) have defined an n-dimensional balanced hypercube, denoted by BH_n, as follows.

Definition 1. *An n-dimensional balanced hypercube* BH_n *consists of* 2^{2n} *nodes* $\left(a_0, a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_{n-1} \right)$, *where* $a_0 \in \left\{ 0, 1, 2, 3 \right\}$ $\left(0 \le i \le n-1 \right)$. *Every node* $\left(a_0, a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_{n-1} \right)$ *connects the following* $2n$ *nodes:*

1. $\left(a_0 + 1, a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_{n-1} \right)$, $\left(a_0 - 1, a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_{n-1} \right)$, *and*

11 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/article/how-to-systematically-embed-cycles-in-balanced-hypercubes/169917

Related Content

A Proposed Pragmatic Software Development Process Model
Sanjay Misra, M. Omorodion, Amit Mishraand Luis Fernandez (2015). *Handbook of Research on Innovations in Systems and Software Engineering (pp. 186-200).*
www.irma-international.org/chapter/a-proposed-pragmatic-software-development-process-model/117925

Investigating the Use of Web 2.0 Technologies and Their Presence in Saudi Government Agencies' Websites
Mohammed Altayar (2018). *Application Development and Design: Concepts, Methodologies, Tools, and Applications (pp. 1371-1390).*
www.irma-international.org/chapter/investigating-the-use-of-web-20-technologies-and-their-presence-in-saudi-government-agencies-websites/188260

Measurement-Based Timing Analysis for Reconfigurable Embedded Systems
Raimund Kirner, Sven Bünteand Michael Zolda (2011). *Reconfigurable Embedded Control Systems: Applications for Flexibility and Agility (pp. 110-129).*
www.irma-international.org/chapter/measurement-based-timing-analysis-reconfigurable/50427

A 3D Chaotic Dynamics-Assisted Color Image Authentication Technique in Multicore Milieu: Multicore Implementation of 3D RGB Steganography
Gaurav Gambhir, Monika Gambhirand Jyotsna Kumar Mandal (2022). *International Journal of Software Innovation (pp. 1-14).*
www.irma-international.org/article/a-3d-chaotic-dynamics-assisted-color-image-authentication-technique-in-multicore-milieu/303581

Model-Driven Software Refactoring
Tom Mens, Gabriele Taentzerand Dirk Müller (2009). *Model-Driven Software Development: Integrating Quality Assurance (pp. 170-203).*
www.irma-international.org/chapter/model-driven-software-refactoring/26830