A Parallel Algorithm for Constructing Two Edge-Disjoint Hamiltonian Cycles in Crossed Cubes

Kung-jui Pai(✉)

Department of Industrial Engineering and Management, Ming Chi University of Technology, New Taipei City, Taiwan
poter@mail.mcut.edu.tw

Abstract. The n-dimensional crossed cube CQ_n, a variation of the hypercube Q_n, has the same number of vertices and the same number of edges as Q_n, but it has only about half of the diameter of Q_n. In the interconnection network, some efficient communication algorithms can be designed based on edge-disjoint Hamiltonian cycles. In addition, two edge-disjoint Hamiltonian cycles also provide the edge-fault tolerant Hamiltonicity for the interconnection network. Hung [Discrete Applied Mathematics 181, 109–122, 2015] designed a recursive algorithm to construct two edge-disjoint Hamiltonian cycles on CQ_n in $O(n2^n)$ time. In this paper, we provide an $O(n)$ time algorithm for each vertex in CQ_n to determine which two edges were used in Hamiltonian cycles 1 and 2, respectively. With the information of each vertex, we can construct two edge-disjoint Hamiltonian cycles in CQ_n with $n \geq 4$.

Keywords: Edge-disjoint Hamiltonian cycles · Crossed cubes · Interconnection networks

1 Introduction

The design of an interconnection network is an important issue for the multicomputer system. The hypercube [17, 18] is one of the most popular interconnection networks because of its attractive properties, including regularity, node symmetric, link symmetric, small diameter, strong connectivity, recursive construction, partition capability, and small link complexity. The architecture of an interconnected network is usually modeled by a graph with vertices representing processing units and edges representing communication links. We will use graph and network interchangeably in this paper.

The n-dimensional crossed cube CQ_n, proposed first by Efe [4, 5], is a variant of an n-dimensional hypercube. One advantage of CQ_n is that the diameter is only about one half of the diameter of an n-dimensional hypercube. Hung et al. [11] showed that CQ_n contains a fault-free Hamiltonian cycle, even if there are up to $2n - 5$ edge faults. Wang studied the embedding of the Hamiltonian cycle in CQ_n [19]. For more properties of CQ_n, the reader can refer to [4–6, 12].

A Hamiltonian cycle is a graph cycle through a graph that visits each node exactly once. The ring structure is important for the multicomputer system, and its benefits can be found in [13]. Two Hamiltonian cycles in the graph are said to be edge-disjoint if they do not share any common edges. Edge-disjoint Hamilton cycles can provide advantages for algorithms using ring structures, and their application can be found in [16]. Further, edge-disjoint Hamiltonian cycles also provide the edge-fault tolerant hamiltonicity for the interconnection network. That is, when one edge in the Hamiltonian cycle fails, the other edge-disjoint Hamiltonian cycle can be adopted to replace it for transmission.

Previous related works are described below. Barth and Raspaud [3] showed that the butterfly networks contain two edge-disjoint Hamiltonian cycles. Bae et al. [1] studied edge-disjoint Hamiltonian cycles in k-ary n-cubes and hypercubes. Then, Barden et al. [2] constructed the maximum number of edge-disjoint spanning trees in a hypercube. Petrovic and Thomassen [15] characterized the number of edge-disjoint Hamiltonian cycles in hypertournaments. Hung presented how to construct two edge-disjoint Hamiltonian cycles in locally twisted cubes [7], augmented cubes [8], twisted cubes [10], transposition networks, and hypercube-like networks [9], respectively. In [9], Hung designed a recursive algorithm to construct two edge-disjoint Hamiltonian cycles in CQ_n. In this paper, we provide a parallel algorithm to construct two edge-disjoint Hamiltonian cycles in CQ_n with $n \geq 4$. Each vertex of CQ_n can simultaneously run this algorithm to know which two edges were used in Hamiltonian cycles 1 and 2, respectively. The recursive algorithm [9] can be adopted by one vertex to constructs two Hamiltonian cycles and this vertex must transfer this message to all other vertices. However, according to our algorithm, each vertex can calculate to get the message, which is more helpful for implementation.

The rest of the paper is organized as follows: In Sect. 2, the structure of crossed cubes is introduced and some notations are given. Section 3 presented two edge-disjoint Hamiltonian cycles in CQ_4. Based on this result, we further show a parallel algorithm to construct two edge-disjoint Hamiltonian cycles in CQ_n with $n \geq 4$. Finally, Sect. 4 is the concluding remarks of this paper.

2 Preliminaries

Interconnection networks are usually modeled as undirected simple graphs $G = (V, E)$, where the vertex set V (=$V(G)$) and the edge set E (=$E(G)$) represent the set of processing units and the set of communication links between nodes, respectively. The neighborhood of a vertex v in a graph G, denoted by $N(v)$, is the set of vertices adjacent to v in G. A cycle C_k of length k in G, denoted by $v_0 - v_1 - v_2 - \ldots - v_{k-2} - v_{k\text{-}1} - v_0$, is a sequence $(v_0, v_1, v_2, \ldots, v_{k-1}, v_0)$ of nodes such that $(v_{k-1}, v_0) \in E$ and $(v_i, v_{i+1}) \in E$ for $0 \leq i \leq k - 2$.

Now, we introduce crossed cubes. A vertex of the n-dimensional crossed cube CQ_n is represented by a binary string of length n. A binary string b of length n is denoted by $b_{n-1}b_{n-2} \cdots\cdots b_1b_0$, where b_{n-1} is the most significant bit. Suppose that G is a labeled graph whose vertices are associated with distinct binary strings, and let G^x be the graph obtained from G by prefixing the binary string on every node with x. Two binary strings $x = x_1x_0$ and $y = y_1y_0$ are pair-related, denoted x–y, if and only if $(x, \in y\{(00, 00), (10,$

10), (01, 11), (11, 01)}. In [5], Efe introduced the notion of pair-related to obtain that the diameter of CQ_n is only about one half of the diameter of Q_n.

Definition 1 (Efe [5].) *The n-dimensional crossed cube CQ_n is the labeled graph with the following recursive fashion:*

1) *CQ_1 is the complete graph on two vertices with labels* 0 *and* 1.

2) *For $n \geq 2$, CQ_n is composed of two subcubes CQ^0_{n-1} and CQ^1_{n-1} such that two vertices $x = 0x_{n-2} \cdots\cdots x_1x_0 \in V(CQ^0_{n-1})$ and $y = 1y_{n-2} \cdots\cdots y_1y_0 \in V(CQ^1_{n-1})$ are joined by an edge if and only if*

(i) $x_{n-2}= y_{n-2}$ *if n is even, and*
(ii) $x_{2i+1}x_{2i}$–$y_{2i+1}y_{2i}$ *for* $0 \leq i < \lfloor (n-1)/2 \rfloor$,

where x and y are called the $(n-1)$*-neighbors to each other, and denote as* $N_{n-1}(x) = y$ *or* $N_{n-1}(y) = x$.

For conciseness, an edge $(x, N_j(x))$ is denoted as $e_j(x)$, and an e_j-edge is an edge $(x, N_j(x))$ in G. Obviously, there are 8 e_3-edges, 8 e_2-edges, 8 e_1-edges and 8 e_0-edges in CQ_4. For example, Fig. 1 shows a 4-dimensional crossed cube CQ_4. A CQ_4 is composed of two subcubes CQ^0_3 (the left half in Fig. 1) *and* CQ^1_3 (the right half in Fig. 1). According to Definition 1 and the notion of pair-related, there exists an e_3-edge connecting vertices 0000 and 1000, and so on. In this paper, sometimes the labels of vertices are changed to their decimal.

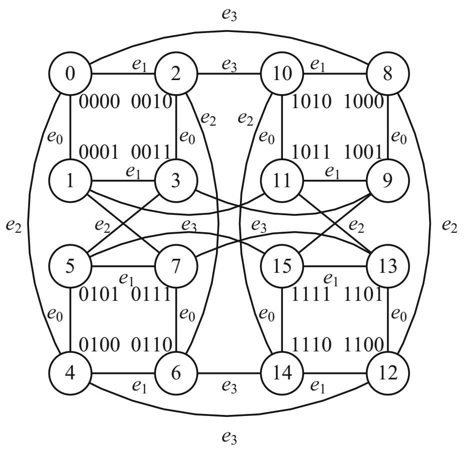

Fig. 1. A 4-dimensional crossed cube CQ_4.

3 Main Results

3.1 Two Edge-Disjoint Hamiltonian Cycles in CQ_4

Hung [9] provided two edge-disjoint Hamiltonian cycles in CQ_4. The first cycle C_{16} is equal to 0100–0000–1000–1001–1111–1101–0111–0101–0011–0001−1011–1010–0010–0110–1110–1100–0100, and there are 6 e_3-edges, 4 e_2-edges, 4 e_1-edges and 2 e_0-edges in it. Since the cycle adopts a different number of edges in each dimension, the parallel construction algorithm that will be presented in the next section becomes more complicated. Fortunately, we found another set of two edge-disjoint Hamiltonian cycles in CQ_4, and each cycle has the same number of edges in each dimension. We described this result in Proposition 2, and its validness can check by Fig. 2.

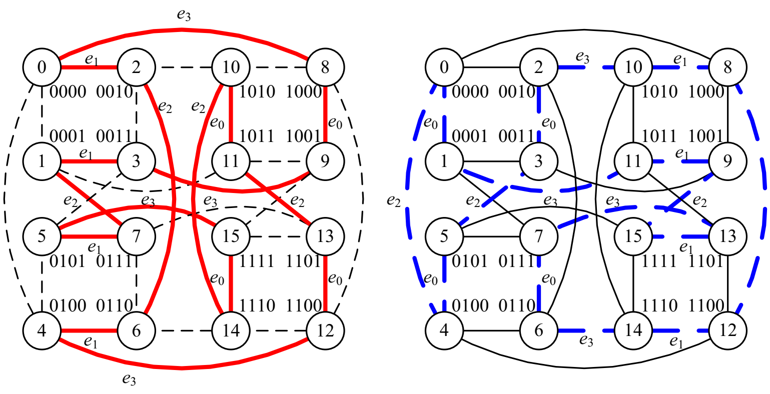

Fig. 2. (a) The first Hamiltonian cycle HC_1 in CQ_4, and (b) the second Hamiltonian cycle HC_2 in CQ_4, where the thick lines indicate the cycle.

Proposition 2. Let HC_1 = 0000–0010–0110–0100–1100–1101–1011–1010–1110–1111–0101–0111–0001–0011–1001–1000–0000, and HC_2 = 0000–0001–1011–1001–1111–1101–0111–0110–1110–1100–1000–1010–0010–0011–0101–0100–0000. HC_1 and HC_2 form two edge-disjoint Hamiltonian cycles in CQ_4.

3.2 Constructing Two Edge-Disjoint Hamiltonian Cycles in CQ_4

Based on the previous results, we now design an algorithm called Algorithm 2HCBase to construct two edge-disjoint Hamiltonian cycles in CQ_4. Each vertex (processing unit) in CQ_4 calls this algorithm and inputs its label to get which two edges are used in Hamiltonian cycles 1 and 2, respectively.

```
Algorithm 2HCBase
Input: $b_3b_2b_1b_0$ in $B$ //$B$ : the label of this vertex
Output: $H_1$ and $H_2$ //$H_i$ : edge set of the $i$-th Hamiltonian cycle
```

```
step 1. if $b_3 = 0$ then $H_1 \leftarrow \{e_1\}$ else $H_1 \leftarrow \{e_0\}$;
step 2. if $b_3 = 0$ then
step 3.     if $b_2 = 0$ then
step 4.       if $b_1$ xor $b_0 = 0$ then $H_1 \leftarrow H_1 \cup \{e_3\}$ else $H_1 \leftarrow H_1 \cup \{e_2\}$;
step 5.     else
step 6.       if $b_1 = 0$ then $H_1 \leftarrow H_1 \cup \{e_3\}$ else $H_1 \leftarrow H_1 \cup \{e_2\}$;
step 7.     end if
step 8. else
step 9.     if $b_2 = 0$ then
step 10.      if $b_1 = 0$ then $H_1 \leftarrow H_1 \cup \{e_3\}$ else $H_1 \leftarrow H_1 \cup \{e_2\}$;
step 11.    else
step 12.      if $b_1$ xor $b_0 = 0$ then $H_1 \leftarrow H_1 \cup \{e_3\}$ else $H_1 \leftarrow H_1 \cup \{e_2\}$;
step 13.    end if
step 14. end if
step 15. $H_2 \leftarrow \{e_3, e_2, e_1, e_0\} \setminus H_1$;
```

In Algorithm 2HCBase, each vertex determines which two edges are used in the first Hamiltonian cycles H_1. In step 1, either e_0-edge or e_1-edge will be selected into H_1. Then, it will add e_2-edge or e_3-edge into H_1 according to steps 2 to 14. Finally, the remaining two edges will be adopted in the second Hamiltonian cycle. Since there is no loop in Algorithm 2HCBase, we have the following lemma.

Lemma 3. The time complexity of Algorithm 2HCBase is $O(1)$.

Lemma 4. *By inputting the label of each vertex into Algorithm 2HCBase, we can obtain 2 cycles, which form two edge-disjoint Hamiltonian cycles in CQ_4.*

Proof. According to step 1 in the algorithm, each vertex selects either e_0-edge or e_1-edge into H_1 by b_0. Then, we provide the decision tree shown in Fig. 3 to illustrate steps 2 through 14 in the algorithm. The two edges selected for each vertex can be checked by Fig. 2. □

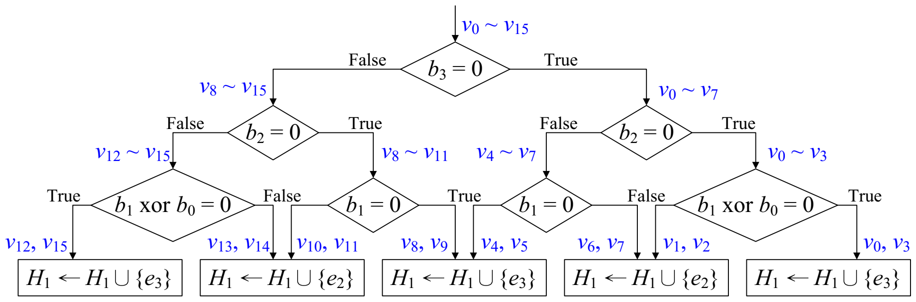

Fig. 3. A decision tree to illustrate steps 2 through 14 in the Algorithm 2HCBase, where v_i represents the vertex i in CQ_4.

3.3 Constructing Two Edge-Disjoint Hamiltonian Cycles in CQ_n for $n \geq 5$

We all know that CQ_{n+1} is composed of CQ_n^0 and CQ_n^1. As described in the previous subsection, there exist two edge-disjoint Hamiltonian cycles in CQ_4. The construction method of Hamiltonian cycle in CQ_{n+1} is as follows. First, we have the Hamiltonian cycle HC_A in CQ_n^0 and the Hamiltonian cycle HC_B in CQ_n^1. Second, remove an edge in HC_A (respectively, HC_B) to get the Hamiltonian path HP_A (respectively, HP_B). Third, connect one end vertex of HP_A and one end vertex of HP_B with an e_n-edge. Finally, connect the other end vertex of HP_A and the other end vertex of HP_B to obtain a Hamiltonian cycle in CQ_{n+1}. Figure 4 illustrates the construction of two such edge-disjoint Hamiltonian cycles in CQ_{n+1}. Base on this method, we design Algorithm 2HC as shown below.

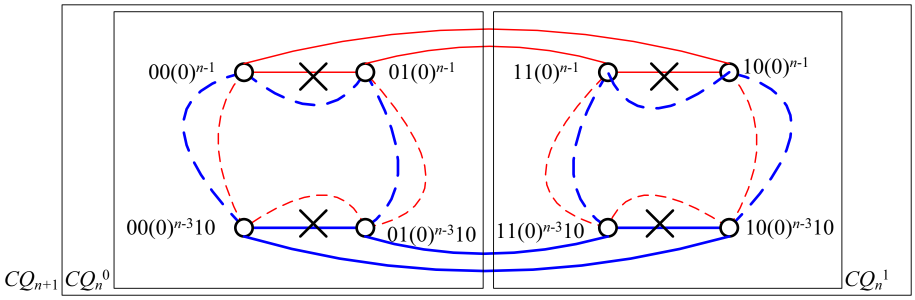

Fig. 4. The construction of two edge-disjoint Hamiltonian cycles in CQ_{n+1} while $n \geq 4$, where thin red lines (respectively, thick blue lines) indicate the first (respectively, the second) Hamiltonian cycle.

```
Algorithm 2HC
Input: $B(= b_{n-1}b_{n-2}\cdots\cdots b_1b_0)$ and $n$ //$B$ : the label of this vertex,
$n$ : the dimension
Output: $HC_1$ and $HC_2$ //$HC_i$ : edge set of the $i$-th Hamiltonian cycle
step 1. By calling Algorithm 2HCBase, $HC_1 \leftarrow H_1$ and $HC_2 \leftarrow H_2$;
step 2. if $b_2 = 0$ and $b_1 = 0$ and $b_0 = 0$ then
step 3.      flag $\leftarrow$ 1
step 4.      for $i = 3$ to $n - 3$ do
step 5.        if $b_i = 1$ then
step 6.          $HC_1 \leftarrow HC_1 \setminus \{e_3\} \cup \{e_{i+1}\}$; flag $\leftarrow$ 0; break for;
step 7.        end if
step 8.      end for
step 9.      if flag $= 1$ then $HC_1 \leftarrow HC_1 \setminus \{e_3\} \cup \{e_{n-1}\}$;
step 10. end if //end if at step 2
step 11. if $b_2 = 0$ and $b_1 = 1$ and $b_0 = 0$ then
step 12.      flag $\leftarrow$ 1
step 13.      for $i = 3$ to $n - 3$ do
step 14.          if $b_i = 1$ then
step 15.          $HC_2 \leftarrow HC_2 \setminus \{e_2\} \cup \{e_{i+1}\}$; flag $\leftarrow$ 0; break for;
step 16.          end if
step 17.       end for
step 18.       if flag = 1 then $HC_2 \leftarrow HC_2 \setminus \{e_2\} \cup \{e_{n-1}\}$;
step 19.  end if //end if at step 11
```

In Algorithm 2HC, each vertex will call Algorithm 2HCBase to obtain HC_1 and HC_2, where HC_i is the edge set of the *i*-th Hamiltonian cycle. In steps 2 to 10, if this vertex is one end vertex of the Hamiltonian path, it will modify its HC_1. For example, in CQ_6, vertex 000000 obtains $HC_1 = \{e_1, e_3\}$ by call Algorithm 2HCBase. Since vertex 000000 is one end vertex of the Hamiltonian path, $HC_1 = \{e_1, e_3\} \setminus \{e_3\} \cup \{e_5\}$. By the same way, in steps 11 to 19, if this vertex is one end vertex of Hamiltonian path, it will modify its HC_2. For example, in CQ_6, vertex 000010 obtains $HC_2 = \{e_1, e_2\}$ by call Algorithm 2HCBase. Since vertex 000010 is one end vertex of the Hamiltonian path, $HC_2 = \{e_1, e_2\} \setminus \{e_2\} \cup \{e_5\}$. According to Lemma 3 and steps 4, 13 in Algorithm 2HC, we have the following lemma.

Lemma 5. The time complexity of Algorithm 2HC is $O(n)$.

Theorem 6. *By inputting the label of each vertex into Algorithm 2HC, we can obtain 2 cycles, which form two edge-disjoint Hamiltonian cycles in CQ_n with $n \geq 4$ in $O(n2^n)$ time. In particular, it can be parallelized on CQ_n to run in $O(n)$ time.*

Proof. Each vertex of CQ_n can simultaneously run Algorithm 2HC to know which two edges were used in Hamiltonian cycles 1 and 2, respectively. By lemma 5, it can be parallelized on CQ_n to run in $O(n)$ time.

Without loss of generality, we consider vertex 0 to be the starting vertex in CQ_n. By Algorithm 2HC, vertex 0 obtains two edges used in the Hamiltonian cycle. With the dimension of the edge, we can obtain the next vertex of the Hamiltonian cycle. In the same way, the vertex sequence of the Hamiltonian cycle can be get. Since there are 2^n vertices in CQ_n, two edge-disjoint Hamiltonian cycles in CQ_n can be obtained in $O(n2^n)$ time. □

For the convenience of checking the correctness of all results, we provide an interactive verification at the website [14]. It shows the usefulness and efficiency of our algorithms in practical settings.

4 Concluding Remarks

In this paper, we first present two edge-disjoint Hamiltonian cycles in CQ_4, and each cycle has the same number of edges in each dimension. Then, we provide an $O(n)$ time algorithm for each vertex in CQ_n to determine which two edges were used in Hamiltonian cycles 1 and 2, respectively. It is interesting to see if there are three edge-disjoint Hamiltonian cycles in CQ_n for $n \geq 6$. So far it is still an open problem.

Acknowledgments. This research was partially supported by MOST grants 107-2221-E-131-011 from the Ministry of Science and Technology, Taiwan.

References

1. Bae, M.M., Bose, B.: Edge disjoint Hamiltonian cycles in k-ary n-cubes and hypercubes. IEEE Trans. Comput. **52**, 1271–1284 (2003)
2. Barden, B., Libeskind-Hadas, R., Davis, J., Williams, W.: On edge-disjoint spanning trees in hypercubes. Inf. Process. Lett. **70**, 13–16 (1999)
3. Barth, D., Raspaud, A.: Two edge-disjoint Hamiltonian cycles in the butterfly graph. Inf. Process. Lett. **51**, 175–179 (1994)
4. Efe, K.: A variation on the hypercube with lower diameter. IEEE Trans. Comput. **40**, 1312–1316 (1991)
5. Efe, K.: The crossed cube architecture for parallel computing. IEEE Trans. Parallel Distrib. Syst. **3**, 513–524 (1992)
6. Efe, K., Blackwell, P.K., Slough, W., Shiau, T.: Topological properties of the crossed cube architecture. Parallel Comput. **20**, 1763–1775 (1994)
7. Hung, R.W.: Embedding two edge-disjoint Hamiltonian cycles into locally twisted cubes. Theor. Comput. Sci. **412**, 4747–4753 (2011)
8. Hung, R.W.: Constructing two edge-disjoint Hamiltonian cycles and two-equal path cover in augmented cubes. J. Comput. Sci. **39**, 42–49 (2012)
9. Hung, R.W.: The property of edge-disjoint Hamiltonian cycles in transposition networks and hypercube-like networks. Discrete Appl. Math. **181**, 109–122 (2015)
10. Hung, R.W., Chan, S.J., Liao, C.C.: Embedding two edge-disjoint Hamiltonian cycles and two equal node-disjoint cycles into twisted cubes. Lect. Notes Eng. Comput. Sci. **2195**, 362–367 (2012)
11. Hung, H.S., Fu, J.S., Chen, G.H.: Fault-free Hamiltonian cycles in crossed cubes with conditional link faults. Inf. Sci. **177**, 5664–5674 (2007)
12. Kulasinghe, P.D., Bettayeb, S.: Multiplytwisted hypercube with 5 or more dimensions is not vertex transitive. Inf. Process. Lett. **53**, 33–36 (1995)
13. Lin, T.J., Hsieh, S.Y., Juan, J.S.-T.: Embedding cycles and paths in product networks and their applications to multiprocessor systems. IEEE Trans. Parallel Distrib. Syst. **23**, 1081–1089 (2012)
14. Pai, K.J.: An interactive verification of constructing two edge-disjoint Hamiltonian cycles in crossed cubes (2020). http://poterp.iem.mcut.edu.tw/2HC_in_CQn/
15. Petrovic, V., Thomassen, C.: Edge-disjoint Hamiltonian cycles in hypertournaments. J. Graph Theory **51**, 49–52 (2006)
16. Rowley, R., Bose, B.: Edge-disjoint Hamiltonian cycles in de Bruijn networks. In: Proceedings of the 6th Distributed Memory Computing Conference, pp. 707–709 (1991)
17. Saad, Y., Schultz, M.H.: Topological properties of hypercubes. IEEE Trans. Comput. **37**, 867–872 (1988)
18. Wang, D.: A low-cost fault-tolerant structure for the hypercube. J. Supercomput. **20**, 203–216 (2001)
19. Wang, D.: On embedding Hamiltonian cycles in crossed cubes. IEEE Trans. Parallel Distrib. Syst. **19**, 334–346 (2008)