The Journal of Supercomputing (2023) 79:4126–4145
https://doi.org/10.1007/s11227-022-04825-5

Three edge-disjoint Hamiltonian cycles in crossed cubes with applications to fault-tolerant data broadcasting

Kung-Jui Pai[1] · Ro-Yu Wu[2] · Sheng-Lung Peng[3] · Jou-Ming Chang[4]

Accepted: 7 September 2022 / Published online: 22 September 2022

Abstract
Multiple edge-disjoint Hamiltonian cycles (EDHCs) provide the advantages of data broadcast in parallel and edge fault-tolerance in network communications. This paper investigates how to construct more EDHCs in a hypercube-variant network called crossed cube, denoted as CQ_n. The topology of CQ_n has more wealth than normal hypercubes in network properties, e.g., it has about half of the diameter of a hypercube with the same dimension. Then, we obtain the following results in this paper: (1) We first provide the construction of three EDHCs in CQ_6. (2) According to the recursive structure of CQ_n, we prove by induction that there exist also three EDHCs in CQ_n for $n \geqslant 7$. (3) Finally, we evaluate the performance of data broadcasting by simulation through three EDHCs and compare it against the best previous result in [18] using two EDHCs. In particular, our findings significantly improved the average success rate in edge fault-tolerant data broadcasting and two specific metrics concerning the broadcasting delivery time (latency).

Keywords Interconnection networks · Edge-disjoint Hamiltonian cycles · Crossed cubes · Fault-tolerant data broadcasting

1 Introduction

The ring structure is essential for high-performance computing architecture, which is often used as the baseband for data transmission in interconnection networks and control flow in parallel and distributed environments. Many efficient algorithms with low communication costs have been developed based on the ring structure [1–4]. To acquire a ring structure passing through every node in a network, it relies on Hamiltonian cycles in the corresponding network topology. In particular, edge-disjoint

A preliminary version of this paper was presented at 23rd International Conference on High Performance Computing and Communications (HPCC 2021), Dec. 20-22, Haikou, Hainan, China.

✉ Jou-Ming Chang
spade@ntub.edu.tw

Extended author information available on the last page of the article

Hamiltonian cycles (EDHCs for short) can provide advantages in many applications, as described below.

Consider the all-to-all communication in a network system with n nodes, where every node needs to send a distinct message to all other nodes. Applying a ring structure, once a message starts sending, a node can receive a new message from the previous node at each step, keeps a copy of the new message for itself, and sends the received message to the next node. Thus, the process can be done through $n-1$ steps. Suppose that the system now contains multiple EDHCs. Then, a message can be partitioned into small packets (e.g., called "frames" in Ethernet), each of which can be sent along the multiple EDHCs. Therefore, the time required for transmission can be reduced. That is, adopting multiple EDHCs as broadband channels in a network can provide multiple data signals/streams simultaneously at the same time in parallel and provide advantages of data transmission. Furthermore, one way to achieve fault-tolerant interprocessor communication is by effectively utilizing disjoint paths between source and destination node pairs. Especially when considering link failure tolerance, the technique of using edge-disjoint paths is a common strategy. Thus, if a fault occurs on one edge of a Hamiltonian cycle, the message can be transmitted along another Hamiltonian cycle. Accordingly, constructing multiple EDHCs has applications to enhance edge fault-tolerant capability in data transmission [5–7].

Networks are usually modeled as undirected simple graphs $G = (V, E)$, where the node set $V(= V(G))$ and the edge set $E(= E(G))$ represent the set of processors and the set of communication channels between processors, respectively. A cycle in a graph that contains every node exactly once is called a *Hamiltonian cycle*. A graph is *Hamiltonian* if it possesses a Hamiltonian cycle. Two cycles in a graph are *edge-disjoint* (resp. node-disjoint) if they share no common edge (resp. common node). It is well-known that finding a Hamiltonian cycle in a graph is an NP-hard problem [8], let alone the problem of finding multiple EDHCs.

Some previous works related to the construction of multiple EDHCs in specific networks are described below. The known existence of two EDHCs include butterfly networks [9], Gaussian networks [10], locally twisted cubes [11, 12], balanced hypercubes [13], augmented cubes [14], twisted cubes [15], parity cubes [16], crossed cubes [17, 18], and some kinds of hypercube-like networks [18] etc. However, so far, only a few studies have been devoted to constructing more than two EDHCs on specific networks. Hussain et al. [19] gave a construction of three EDHCs in Eisenstein-Jacobi networks. Hung [18] showed that the n-dimensional transposition networks for $n \geqslant 5$ contains four EDHCs. For concerning the existence of more EDHCs in graphs or networks, please refer to [20] for hypertournaments, [21] for WK-recursive networks, [5] for hypercubes and k-ary n-cubes, and [22] for generalized hypercubes.

This paper studies the problem of constructing three EDHCs in a class of hypercube-variant networks called crossed cubes and denoted by CQ_n (see Definition 1). As constructing beyond two EDHCs in CQ_n for $n \geqslant 6$ is currently an open problem, our study of demonstrating the existence of three EDHCs on CQ_n is a theoretical breakthrough. This result improves the previous ones of Pai [17] and Hung [18] in terms of quantity of Hamiltonian cycles, where the former first designed a recursive algorithm to construct two EDHCs of CQ_n in $\mathcal{O}(n2^n)$ time when $n \geqslant 4$, and the latter

presented an elegant parallel construction to yield two EDHCs in $\mathcal{O}(n)$ time using 2^n nodes of CQ_n as processors. Using the three EDHCs in CQ_n as the transmission channels, we can easily carry out an edge fault-tolerant broadcast from an arbitrary node. In addition, using three EDHCs instead of two EDHCs can improve the performance of fault-tolerant and latency in data broadcasting.

So far, the Hamiltonicity of the crossed cube has been studied extensively in the literature. For instance, research on fault-tolerant Hamiltonicity (or pancyclicity) [23–26], Hamiltonicity with conditional link faults [27, 28], and Hamiltonicity (or panconnectedness) with required nodes in the fixed positions [29, 30]. Excepting the above research, the recent issues that have paid more attention to crossed cubes are the construction of multiple spanning trees [31–38], node-to-set disjoint paths [39], and the study of (conditional) diagnosability [40, 41]. For more properties and results of CQ_n, the reader can refer to [42, 43].

The rest of the paper is organized as follows. Section 2 formally gives the definition of crossed cubes and introduces a previous result showing two EDHCs of CQ_4. Section 3 presents how to construct three EDHCs of CQ_6. Section 4 provides a recursive algorithm to construct three EDHCs of CQ_n for $n \geqslant 7$. Section 5 simulates edge fault-tolerant data broadcasting in CQ_n for $6 \leqslant n \leqslant 10$ and compares the experimental performance with the best previous results. Finally, the concluding remarks are given in the last section.

2 Preliminaries

Suppose that G is a labeled graph whose nodes 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. Efe [44] defined two binary strings $x = x_1x_0$ and $y = y_1y_0$ to be *pair-related*, denoted $x \sim y$, if and only if $(x, y) \in \{(00, 00), (10, 10), (01, 11), (11, 01)\}$.

Definition 1 (Efe [44].) 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 nodes with labels 0 and 1.

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

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

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 of representation, sometimes the labels of nodes are changed to the use of decimal. For instance, Fig. 1 shows two EDHCs of the 4-dimensional

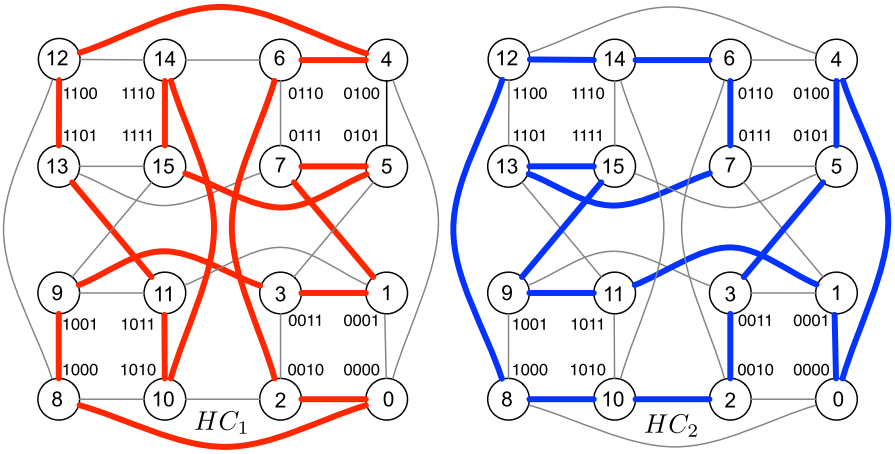

Fig. 1 Two EDHCs HC_1 and HC_2 in CQ_4, where thick lines indicate edges of cycles

crossed cube CQ_4 that was provided in [17], where each node is labeled by the binary code and its corresponding decimal (inside the circle). We can see that each Hamiltonian cycle in HC_1 and HC_2 contains 16 distinct edges, of which eight edges are evenly distributed in four subcubes CQ_2^{00}, CQ_2^{01}, CQ_2^{10}, and CQ_2^{11}, and then through eight different outer edges are joined to form a Hamiltonian cycle of CQ_4.

3 Three edge-disjoint Hamiltonian cycles in CQ_6

In this section, we first give two EDHCs of CQ_6 that were built from [17]. Then, we fine-tune some edges in the second Hamiltonian cycle so that the remaining edges are sufficient to construct the third Hamiltonian cycle.

A cycle with length ℓ is said to be an *ℓ-cycle*. Let C be a cycle. For notational convenience, we write $e \in C$ instead of $e \in E(C)$. Let HC_1 and HC_2 be two EDHCs of CQ_4 (see Fig. 1). By the recursive structure in Definition 1, CQ_6 can be decomposed into four copies of CQ_4, and each copy has two Hamiltonian cycles mentioned above. Hence, Pai [17] provided the following way to construct two Hamiltonian cycles of CQ_6. Partition CQ_6 into four disjoint subcubes CQ_4^{ij} for $i,j \in \{0,1\}$, and let HC_k^{ij} for $k \in \{1,2\}$ be the corresponding kth Hamiltonian cycle in the subcube CQ_4^{ij} such that each cycle maps to HC_k in CQ_4. For each $k = 1,2$, since each subcube contains Hamiltonian cycles passing through distinct nodes, the four Hamiltonian cycles in the subcubes are easily constructed in a parallel fashion. Then, the two Hamiltonian cycles of CQ_6, say $\mathcal{HC}_1$ and $\mathcal{HC}_2$, can be constructed from the merge of HC_k^{ij} for $k = 1,2$ by adjusting four edges in each cycle, which are described as follows:

$$E(\mathcal{HC}_1) = \left(\bigcup_{i,j\in\{0,1\}} E(HC_1^{ij}) \right) \cup \{(0,32),(8,24),(16,48),(40,56)\} - \{(0,8),(16,24),(32,40),(48,56)\} \quad (1)$$

and

$$E(\mathcal{HC}_2) = \left(\bigcup_{i,j\in\{0,1\}} E(HC_2^{ij}) \right) \cup \{(2,34),(10,26),(18,50),(42,58)\} - \{(2,10),(18,26),(34,42),(50,58)\}. \quad (2)$$

For instance, Fig. 2 depicts the Hamiltonian cycle $\mathcal{HC}_1$ of CQ_6 constructed from Eq. (1). From the drawing, we may imagine that two dashed lines (horizontal and vertical) partition the whole CQ_6 into four subcubes with equal size, where nodes in the upper part and lower part (resp. left part and right part) are mirrored, and their labels have the difference ± 16 (resp. ± 32). Due to saving the space, we omit the drawing of $\mathcal{HC}_2$ constructed from Eq. (2). However, since certain edges are contained in HC_2 (see Fig. 1), we can be sure that some corresponding edges exist in $\mathcal{HC}_2$, e.g., we have the following mapping of edges from HC_2 to $\mathcal{HC}_2$:

$$\begin{aligned}
(8,12) \in HC_2 &\rightarrow (56,60) \in \mathcal{HC}_2 && // + 48\\
(4,5) \in HC_2 &\rightarrow (20,21) \in \mathcal{HC}_2 && // + 16\\
(15,9) \in HC_2 &\rightarrow (63,57) \in \mathcal{HC}_2 && // + 48\\
(11,1) \in HC_2 &\rightarrow (27,17) \in \mathcal{HC}_2 && // + 16\\
(3,5) \in HC_2 &\rightarrow (51,53) \in \mathcal{HC}_2 && // + 48\\
(5,4) \in HC_2 &\rightarrow (\ 5,\ 4) \in \mathcal{HC}_2 && // + 0\\
(12,14) \in HC_2 &\rightarrow (44,46) \in \mathcal{HC}_2 && // + 32\\
(6,14) \in HC_2 &\rightarrow (54,62) \in \mathcal{HC}_2 && // + 48\\
(6,7) \in HC_2 &\rightarrow (38,39) \in \mathcal{HC}_2 && // + 32\\
(13,7) \in HC_2 &\rightarrow (61,55) \in \mathcal{HC}_2 && // + 48\\
(13,15) \in HC_2 &\rightarrow (45,47) \in \mathcal{HC}_2 && // + 32\\
(15,9) \in HC_2 &\rightarrow (47,41) \in \mathcal{HC}_2 && // + 32\\
(11,1) \in HC_2 &\rightarrow (59,49) \in \mathcal{HC}_2 && // + 48\\
(3,2) \in HC_2 &\rightarrow (19,18) \in \mathcal{HC}_2 && // + 16\\
(10,8) \in HC_2 &\rightarrow (26,24) \in \mathcal{HC}_2 && // + 16
\end{aligned}$$

Let A be the set consisting of the above fifteen edges in $\mathcal{HC}_2$. Since CQ_6 contains 192 edges, it has the remaining 64 edges after removing two EDHCs $\mathcal{HC}_1$ and $\mathcal{HC}_2$. With brute force checking, we can find that $E(CQ_6) - \{E(\mathcal{HC}_1) \cup E(\mathcal{HC}_2)\}$ includes two 8-cycles and twelve 4-cycles, and all these cycles are node-disjoint. We list all fourteen cycles in detail as follows:

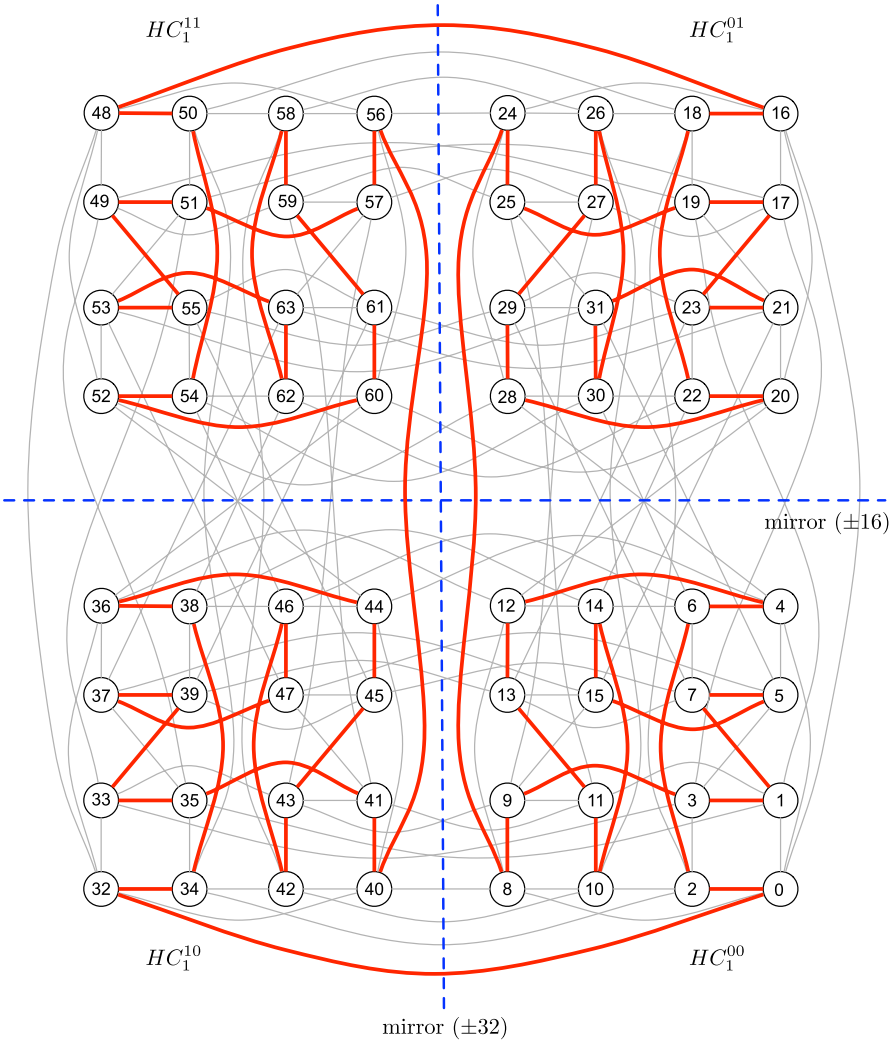

Fig. 2 The Hamiltonian cycle $\mathcal{HC}_1$ of CQ_6 constructed from Eq. (1), where thick lines indicate edges of the cycle

$C_1 = (56, 48, 32, 40, 8, 0, 16, 24, 56)$, $C_2 = (60, 36, 12, 20, 60)$,
$C_3 = (21, 15, 37, 63, 21)$, $C_4 = (57, 43, 9, 27, 57)$,
$C_5 = (17, 3, 33, 51, 17)$, $C_6 = (53, 31, 5, 47, 53)$,
$C_7 = (4, 28, 52, 44, 4)$, $C_8 = (46, 6, 30, 54, 46)$,
$C_9 = (62, 22, 14, 38, 62)$, $C_{10} = (39, 61, 23, 13, 39)$,
$C_{11} = (55, 29, 7, 45, 55)$, $C_{12} = (11, 25, 59, 41, 11)$,
$C_{13} = (35, 1, 19, 49, 35)$, $C_{14} = (2, 10, 42, 34, 50, 58, 26, 18, 2)$.

Let B be the set of edges picking from these cycles:

$$\begin{array}{lll} (24,56) \in C_1, & (20,60) \in C_2, & (63,21) \in C_3, \\ (27,57) \in C_4, & (51,17) \in C_5, & (5,47), (47,53) \in C_6, \\ (44,\ 4) \in C_7, & (54,46) \in C_8, & (38,62) \in C_9, \\ (39,61) \in C_{10}, & (45,55) \in C_{11}, & (59,41) \in C_{12}, \\ (19,49) \in C_{13}, & (26,18) \in C_{14}. & \end{array}$$

Note that we pick two edges of C_6 in B, and thus B contains fifteen edges.

The third Hamiltonian cycle of CQ_6, say $\mathcal{HC}_3$, can be constructed as follows (see Fig. 3 for illustration):

$$E(\mathcal{HC}_3) = \left(\bigcup_{i=1}^{14} E(C_i) - B \right) \cup A. \tag{3}$$

Figure 3 illustrates the construction of $\mathcal{HC}_3$, where the bold lines indicate the edges belonging to A, which will be added to the edge set of $\mathcal{HC}_3$. While lines with cross marks are edges belonging to B, which will be removed from $\mathcal{HC}_3$. In fact, the construction of the third Hamiltonian cycle mentioned above is done using the edge-swapping technique. Therefore, with the above adjustments, we need to modify the edge set of $\mathcal{HC}_2$ obtained from Eq. (2) by swapping two edge sets, A and B, as follows (see Fig. 4 for illustration):

$$E(\mathcal{HC}_2) = (E(\mathcal{HC}_2) - A) \cup B. \tag{4}$$

Similarly, in Fig. 4, the bold lines indicate the edges of B that will be added, while lines with cross marks are edges of A that will be removed in the construction. We now easily check $\{u, v : (u, v) \in A\} = \{u, v : (u, v) \in B\}$, i.e., the unions of ends of edges in the two sets are identical. From an arbitrary starting node, by visiting the paths formed by the edges in Figs. 3 and 4 in sequence, we finally obtain two EDHCs, and hence the result is shown below.

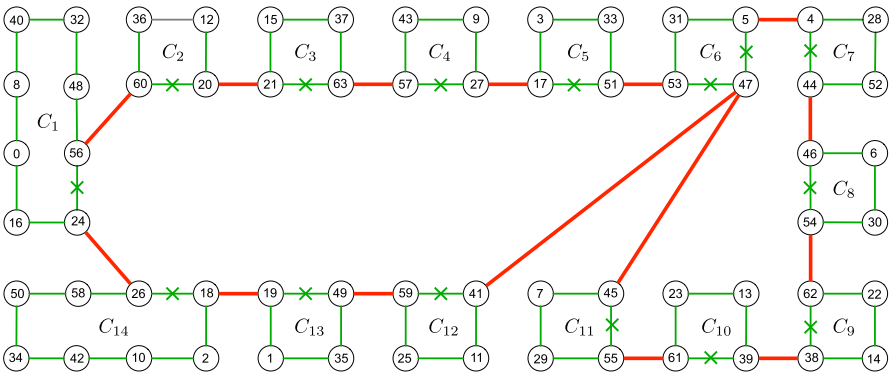

Fig. 3 The Hamiltonian cycle $\mathcal{HC}_3$ of CQ_6 constructed from Eq. (3)

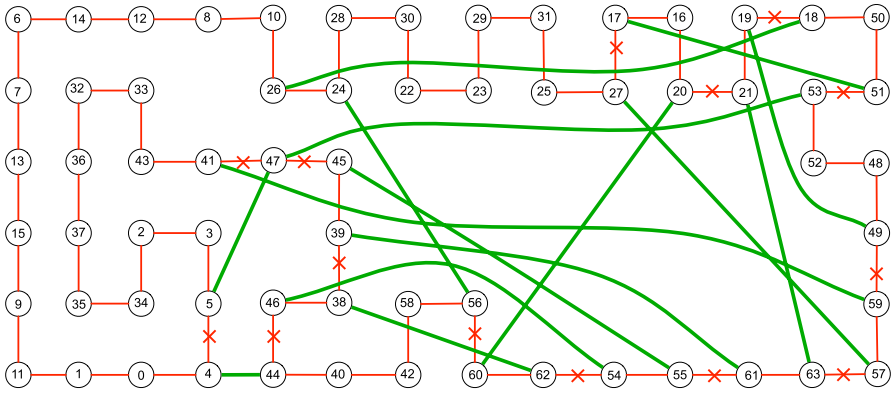

Fig. 4 The Hamiltonian cycle $\mathcal{HC}_2$ of CQ_6 constructed from Eq. (4)

Lemma 1 *The three Hamiltonian cycles $\mathcal{HC}_i$ for $i = 1, 2, 3$ constructed from Eqs.* (1)-(4) *are edge-disjoint in CQ_6.*

Clearly, $|E(CQ_n)| = (nN)/2$, where $N = 2^n$ is the number of nodes of CQ_n. When $n = 6$, we have $N = 2^6 = 64$ and $|E(CQ_6)| = (6 \times 64)/2 = 192$. Since each Hamiltonian cycle contains 64 edges in CQ_6, all edges of CQ_6 will be exhausted when we construct the above three EDHCs. By Eqs. (1)-(4), the time required in the above construction is proportional to the number of edges, and thus it requires $|E(CQ_6)| = 3N = \mathcal{O}(N)$ time.

4 Three edge-disjoint Hamiltonian cycles in CQ_n for $n \geqslant 7$

For constructing three EDHCs in high-dimensional crossed cubes, the results of CQ_6 can be viewed as the induction base, and the construction for CQ_n, $n \geqslant 7$, will be proceeded by recursion. Before this, we need the following prefatory works.

For $n \geqslant 6$, a node $x = x_{n-1}x_{n-2}\cdots x_1x_0$ in CQ_n is said to be a *stable node* if $x_{2i+1}x_{2i} \in \{00, 10\}$ for all $0 \leqslant i < \lfloor n/2 \rfloor$. Let SN_n be the set consisting of all stable nodes of CQ_n. An edge is called a *stable edge* in a subgraph of CQ_n provided its two ends are stable nodes. A Hamiltonian cycle of CQ_n is *sustainable* if it contains a stable edge.

For example, CQ_6 contains eight stable nodes as follows:

$SN_6 = \{\underline{00}\,\underline{00}\,\underline{00}\,(0),\ \underline{00}\,\underline{00}\,\underline{10}\,(2),\ \underline{00}\,\underline{10}\,\underline{00}\,(8),\ \underline{00}\,\underline{10}\,\underline{10}\,(10),$

$\underline{10}\,\underline{00}\,\underline{00}\,(32),\ \underline{10}\,\underline{00}\,\underline{10}\,(34),\ \underline{10}\,\underline{10}\,\underline{00}\,(40),\ \underline{10}\,\underline{10}\,\underline{10}\,(42)\}$.

Let us check Figs. 2, 4, and 3 to realize whether the three EDHCs obtained in the previous section for CQ_6 are sustainable. Since $(0, 2) \in \mathcal{HC}_1$, $(8, 10) \in \mathcal{HC}_2$, and $(32, 40) \in \mathcal{HC}_3$, all three Hamiltonian cycles are indeed sustainable.

For $n \geqslant 7$ and $j \in \{0,1\}$, let $SN_n^j = \{jx : x \in SN_{n-1}\}$. Clearly, from the definition of stable nodes, we have the following recursion.

$$SN_n = \begin{cases} SN_{n-1}^0 \cup SN_{n-1}^1 & \text{if } n \geqslant 7 \text{ is odd;} \\ \{jx : x \in SN_{n-2}^0, j \in \{0,1\}\} & \text{if } n \geqslant 8 \text{ is even.} \end{cases} \tag{5}$$

Lemma 2 *For $n \geqslant 7$, if $x \in SN_{n-1}$, then $(0x, 1x) \in E(CQ_n)$.*

Proof Let $x = x_{n-2}x_{n-3}\cdots x_1x_0$ be a stable node of CQ_{n-1}. Then, $x_{2i+1}x_{2i} \in \{00, 10\}$ for $0 \leqslant i < \lfloor (n-1)/2 \rfloor$. Clearly, $0x\,(= 0x_{n-2}x_{n-3}\cdots x_1x_0) \in V(CQ_{n-1}^0)$ and $1x\,(= 1x_{n-2}x_{n-3}\cdots x_1x_0) \in V(CQ_{n-1}^1)$. Since the two nodes $0x$ and $1x$ have different only at the leftmost bit, by the pair-related property of crossed cubes, it follows from Definition 1 that $(0x, 1x) \in E(CQ_n)$. □

Suppose that we have constructed three EDHCs HC_i for all $i = 1, 2, 3$ in CQ_{n-1}, where $n \geqslant 7$. In what follows, we use Algorithm 1, taking HC_i as the input, to construct the required Hamiltonian cycle $\mathcal{HC}_i$ for each $i \in \{1, 2, 3\}$ in CQ_n. For notational convenience, we omit the subscript i of the Hamiltonian cycle (see Fig. 5 for illustration.)

We now show that Hamiltonian cycles described in Algorithm 1 have the following properties.

Algorithm 1: HC-Construction (HC)

Input: A Hamiltonian cycles HC with a stable edge (x, y) in CQ_{n-1}.
Output: A Hamiltonian cycles $\mathcal{HC}$ with a stable edge in CQ_n.

1 **for** $j \in \{0,1\}$ **do**
2 Let HC^j be the corresponding Hamiltonian cycle of CQ_{n-1}^j that maps HC in CQ_{n-1};
3 Let $e_j = (jx, jy)$ be the edge in HC^j;
4 $E(\mathcal{HC}) = E(HC^0) \cup E(HC^1) \cup \{(0x, 1x), (0y, 1y)\} - \{e_0, e_1\}$;
5 **return** $\mathcal{HC}$ and its stable edge $(0x, 1x)$.

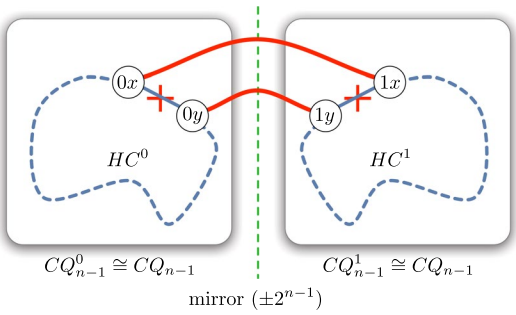

Fig. 5 A Hamiltonian cycle $\mathcal{HC}$ of CQ_n constructed from Algorithm 1

Lemma 3 *For odd $n \geqslant 7$, if the Hamiltonian cycle HC of CQ_{n-1} is sustainable, then so is $\mathcal{HC}$ of CQ_n.*

Proof Let $(x, y) \in HC$ be a stable edge in the Hamiltonian cycle HC of CQ_{n-1}. Since $x, y \in SN_{n-1}$, Lemma 2 shows the existence of edges $(0x, 1x)$, $(0y, 1y) \in E(CQ_n)$, where $0x, 0y \in SN^0_{n-1}$ and $1x, 1y \in SN^1_{n-1}$. Since $n \geqslant 7$ is odd, by Eq. (5), it follows that $0x, 0y, 1x, 1y \in SN_n$. Thus, by the construction of Algorithm 1 (see Line 4), the two edges $(0x, 1x), (0y, 1y) \in \mathcal{HC}$ are stable edges. □

Lemma 4 *For even $n \geqslant 8$, if the Hamiltonian cycle HC of CQ_{n-1} is sustainable, then so is $\mathcal{HC}$ of CQ_n.*

Proof Since $(n-1) \geqslant 7$ is odd, by Lemma 3, the Hamiltonian cycle HC constructed by Algorithm 1 in CQ_{n-1} contains a specific stable edge $(0x, 1x)$, where $x \in SN_{n-2}$. Let $x' = 0x$ and $y' = 1x$. Since $x \in SN_{n-2}$, we have $x'(= 0x) \in SN^0_{n-2}$ and $y'(= 1x) \in SN^1_{n-2}$. Since $n-1$ is odd, it follows $x', y' \in SN_{n-1}$. By Lemma 2, $(0x', 1x'), (0y', 1y') \in E(CQ_n)$. Also, since $x' \in SN^0_{n-2}$ and $n \geqslant 8$ is even, by Eq. (5), it follows that $0x', 1x' \in SN_n$. Thus, by the construction of Algorithm 1 (see Line 4), the edge $(0x', 1x') \in \mathcal{HC}$ is a stable edge. □

The three sustainably EDHCs of CQ_6 are the base of the recursive construction. The correctness of constructing sustainably Hamiltonian cycles for high-dimensional crossed cubes in Algorithm 1 is directly followed from the induction rules described in Lemmas 3 and 4. The edge-disjoint property is evident because the distinct Hamiltonian cycles $\mathcal{HC}_i$ obtained in CQ_n are derived from the EDHCs HC_i of CQ_{n-1}. Also, the construction of each Hamiltonian cycle is in a linear time. Therefore, we have the following main result.

Theorem 1 *For $n \geqslant 6$, there exist three edge-disjoint Hamiltonian cycles in CQ_n. In particular, each Hamiltonian cycle can be constructed in $\mathcal{O}(N)$ time, where $N = 2^n$ is the number of nodes in CQ_n.*

5 Simulation of data broadcasting using multiple EDHCs

This section aims to simulate data broadcasting using multiple EDHCs as the transmission channels in crossed cubes. We compare our results against those of Hung [18] by analyzing the performance using three or two EDHCs. Particularly, we evaluate the capability of fault tolerance by the average success rate in broadcasting with multiple edge failures. In addition, we also assess the efficiency through two metrics related to the *broadcasting delivery time* (or call the *broadcasting latency*) in the experiment.

The networks dealt with by the simulation are CQ_n for $n \in \{6, 7, 8, 9, 10\}$. Technically, algorithms of constructing EDHCs and of simulating data broadcasting are

implemented using C programs. We proceed the experiment by using a 3.60GHz Intel®Core™ i7-8700 CPU and 8 GB RAM under the Linux operating system.

5.1 Evaluation of average success rate in fault-tolerant data broadcasting

Firstly, we are interested in evaluating the *average success rate* (ASR for short), which is the ratio of the number of successful data broadcasts over generated instances. Let F be the set of faulty edges and the number of edge in F considered in this section is within the range between 1 and 10. Here we carry out 10^8 simulation instances for each situation. The source node of broadcasting and the set of faulty edges are randomly generated by the uniform distribution over the whole network CQ_n for $6 \leqslant n \leqslant 10$.

Tables 1 and 2 show the simulation results of ASR for data broadcasting in crossed cubes CQ_n adopting Hung's two EDHCs [18] and our three EDHCs as the broadcasting channels, respectively.

In addition, two quantities, namely *hit rate* (HR for short) and *density of non-Hamiltonian edges* (DnHe for short) are compared (see the two rightmost columns in the tables). The former refers to the ratio of randomly generated edge failures that occur on non-Hamiltonian cycles, and the latter means the ratio of the edges of non-Hamiltonian cycles to the entire network. Let t be the number of EDHCs simulated in the experiment, and let e_j denote the faulty edge in the jth simulation (where $1 \leqslant j \leqslant 10^8$) when $|F| = 1$. Specifically, for a certain dimension of crossed cubes CQ_n, we use the following formula to compute the hit rate:

$$\mathrm{HR} = \sum_{j=1}^{10^8} \frac{\prod_{i=1}^{t}(1 - \mathcal{HC}_i(e_j))}{10^8},$$

where $\mathcal{HC}_i(e_j)$ is an indicator, i.e., $\mathcal{HC}_i(e_j) = 1$ if the faulty edge e_j is contained in the ith Hamiltonian cycle, and $\mathcal{HC}_i(e) = 0$ otherwise. Also, the term DnHe is only depending on n and t as follows:

$$\mathrm{DnHe} = \frac{|E(CQ_n)| - \sum_{i=1}^{t} |E(\mathcal{HC}_i)|}{|E(CQ_n)|} = 1 - \frac{t \cdot 2^n}{n \cdot 2^{n-1}} = 1 - \frac{2t}{n}.$$

We observe from Tables 1 and 2 that the two quantities HR and DnHe are almost very near or even the same. Figure 6 shows the corresponding trend of the ASR concerning the scales of CQ_n and the number of faulty edges when $t = 2$ (i.e., Hung's two EDHCs in [18]) and $t = 3$ (i.e., three EDHCs developed in this paper), respectively.

From Tables 1, 2 and Fig. 6, we are aware of the following three phenomenons:

- Because our result (resp. Hung's result in [18]) provides three EDHCs (resp. two EDHCs) as broadcasting channels, no matter what scale of CQ_n for $6 \leqslant n \leqslant 10$, its transmission can reach 100% success when $|F| = 1, 2$ (resp. $|F| = 1$). The intuitive idea is that ASR should present a decreasing function to the number of faulty nodes. As expected, all simulations are consistent with this phenom-

Table 1 ASR of fault-tolerant data broadcasting in crossed cubes using two EDHCs of Hung [18] under the variation of multiple edge failures

|F| ASR CQ_n	1	2	3	4	5	6	7	8	9	10	Hit rate when |F| = 1	The density of non-Hamiltonian edges
CQ_6	1.0000	0.7766	0.5521	0.3773	0.2528	0.1675	0.1104	0.0723	0.0471	0.0307	0.333332	64/192 = 0.3333
CQ_7	1.0000	0.8364	0.6489	0.4847	0.3546	0.2561	0.1837	0.1311	0.0932	0.0662	0.428584	192/448 = 0.4286
CQ_8	1.0000	0.8748	0.7183	0.5694	0.4421	0.3387	0.2575	0.1945	0.1466	0.1100	0.500017	512/1024 = 0.5000
CQ_9	1.0000	0.9012	0.7694	0.6363	0.5159	0.4127	0.3273	0.2579	0.2024	0.1584	0.555606	1280/2304 = 0.5556
CQ_{10}	1.0000	0.9200	0.8079	0.6895	0.5774	0.4773	0.3911	0.3183	0.5279	0.2082	0.599992	3072/5120 = 0.6000

Table 2 ASR of fault-tolerant data broadcasting in crossed cubes using three EDHCs developed in this paper under the variation of multiple edge failures

|F| ASR CQ_n	1	2	3	4	5	6	7	8	9	10	Hit rate when |F| = 1	The density of non-Hamiltonian edges
CQ_6	1.0000	1.0000	0.7742	0.5485	0.3736	0.2496	0.1650	0.1083	0.0707	0.0460	0.000000	0/192=0.0000
CQ_7	1.0000	1.0000	0.8591	0.6777	0.5107	0.3754	0.2718	0.1950	0.1392	0.0991	0.142859	64/448=0.1429
CQ_8	1.0000	1.0000	0.9061	0.7648	0.6177	0.4853	0.3748	0.2862	0.2169	0.1636	0.249964	256/1024=0.2500
CQ_9	1.0000	1.0000	0.9341	0.8242	0.6987	0.5765	0.4669	0.3737	0.2962	0.2335	0.333371	768/2304=0.3333
CQ_{10}	1.0000	1.0000	0.9520	0.8655	0.7598	0.6502	0.5464	0.4530	0.3721	0.3035	0.400015	2048/5120=0.4000

enon. For instance, we take CQ_{10} in Table 2 as an example for illustration. If we ask ASR $\geqslant 80\%$, then it only permits four faulty nodes, while it can tolerate seven faulty nodes when we allow no more than half of the broadcasts to fail. Accordingly, this means that the least number of edge failures, the larger the corresponding ASR.

- For $|F| > 2$, the ASR increases with expanding the scale of CQ_n. The reason is evident because when $|F|$ is fixed, the edge failures that occur in Hamiltonian cycles will reduce their probability as the network expands, and thus leading to an increase in the success rate. Similarly, by the reasoning that the density of non-Hamiltonian edges gradually increases as the network grows, the hit rate also increases with the scale of the network.
- To highlight the difference in the performance of ASR using our three EDHCs and Hung's two EDHCs as broadcasting channels, we take the extreme cases of CQ_6 and CQ_{10} as examples to illustrate. Obviously, the former is always better than the latter when $|F| \geqslant 2$ (see Fig. 7). In the low-dimensional CQ_6, the amplitude of the ASR difference varies more apparent, but the difference will gradually decrease with the increase of $|F|$. In contrast, in high-dimensional CQ_{10}, the magnitude of the ASR difference is relatively stable with the change of $|F|$.

5.2 Evaluation of packet delivery time in data broadcasting

Next, we simulate the scenario that there exists a message of size no more than 5 M from a source node in CQ_n to broadcast through multiple EDHCs. With the most common Ethernet frame, its frame length can carry about 1500 bytes of the message (excluding the initial preamble, frame delimiter, and the frame check sequence at the end). Hence, the message can be divided into at most $5\text{M}/1500\,\text{bytes} = 3500$ (data packets). To comprehend the difference in broadcasting efficiency between two EDHCs (generated by Hung [18]) and three EDHCs

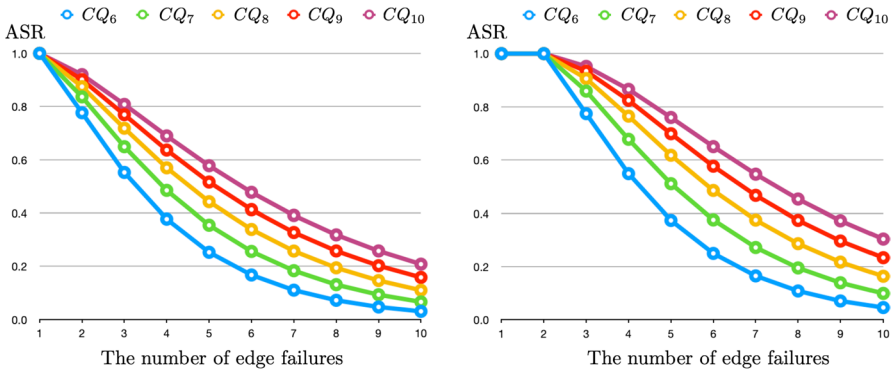

Fig. 6 The trend of the ASR with respect to the scales of CQ_n for $6 \leqslant n \leqslant 10$ and the number of faulty edges $1 \leqslant |F| \leqslant 10$: (a) Simulation using two EDHCs of Hung [18]; (b) Simulation using three EDHCs developed in this paper

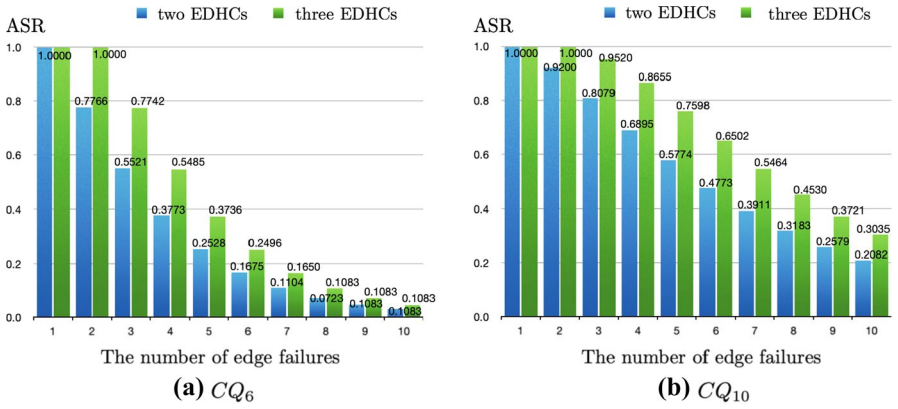

Fig. 7 The comparisons of ASR between our three EDHCs and Hung's two EDHCs as the fault-tolerant broadcasting channels: **a** Simulation in CQ_6; **b** Simulation in CQ_{10}

(developed in this paper), we carry out 10^6 simulation instances for each case. For the sake of fairness, the same message scripts are used for both cases.

For each dimension n with $6 \leqslant n \leqslant 10$, the source node s at each broadcasting is randomly generated by the uniform distribution over the whole network CQ_n. Let m be the number of data packets for broadcasting that is randomly chosen in the range $1 \leqslant m \leqslant 3500$. As the source s has two adjacent nodes in a Hamiltonian cycle, the broadcasting randomly picks a direction to carry out the propagation. A data packet will be sent to the next node sequentially along the Hamiltonian cycle at each time slot until it finally returns to the origin to represent the success of this round of transmission. Suppose we take t EDHCs as transmission channels for data broadcasting (actually, $t = 2$ or $t = 3$ in the following simulation). To balance the load of all transmission channels, we use a round-robin strategy to invoke these channels for packet propagation sequentially. That is, the first t packets are propagated by the channel to which they belong in sequence, the first channel bears the $(t + 1)$-th packet, the second channel bears the $(t + 2)$-th packet..., and so on.

For each node $v \in V(CQ_n) - \{s\}$ and each j with $1 \leqslant j \leqslant m$, let $t_v(j)$ be the delivery time that v receives the jth packet, which is calculated from the beginning. Then, the time for node v receiving the whole message can be represented by $t(v) = \max_{j=1}^{m} \{t_v(j)\}$. For broadcasting the message ms_i (where $1 \leqslant i \leqslant 10^6$) containing m packets, we define two specific metrics called the *average delivery time* and the *maximum delivery time* as follows:

$$\mu(ms_i) = \frac{\sum_{v \in V(CQ_n)-\{s\}} \max_{j=1}^{m} \{t_v(j)\}}{|V(CQ_n)| - 1}$$

and

$$\eta(ms_i) = \max_{v \in V(CQ_n)} \left\{ \max_{j=1}^{m} \{t_v(j)\} \right\}.$$

Specifically, for a certain dimension of crossed cubes CQ_n, we use the following two measures to evaluate broadcasting efficiency, one is called the *average broadcasting latency* (ABL for short), and the other is the *maximum broadcasting latency* (MBL for short):

$$\text{ABL} = \frac{\sum_{i=1}^{10^6} \mu(ms_i)}{10^6}$$

and

$$\text{MBL} = \max_{i=1}^{10^6} \{\eta(ms_i)\}.$$

All experimental results showing ABL and MBL are depicted in Fig. 8. From this figure, we are aware of the following three phenomenons:

- As the network size grows, the two measures ABL and MBL will also rise. That is, both ABL and MBL should exhibit a positive correlation with the network scale.
- For both measures ABL and MBL, the performance of broadcasting latency using three EDHCs is better than that using two EDHCs. For example, the ratio of ABL of three EDHCs to that of two EDHCs approaches 68.85% in CQ_6 and is near 85.54% in CQ_{10}. In contrast, the ratio of MBL of three EDHCs to that of two EDHCs comes to 67.83% in CQ_6 and is near 78.97% in CQ_{10}.

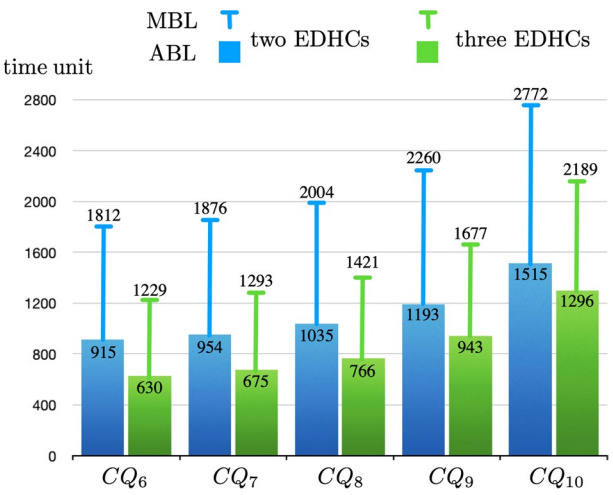

Fig. 8 The comparisons of ABL and MBL between our three EDHCs and Hung's two EDHCs as the broadcasting channels

- Within the network dimension range of our experiments, the value of MBL is approximately twice that of ABL. For example, for CQ_n with n from 6 to 10, the ratios appear in order 1.98, 1.97, 1.94, 1.89, and 1.83 for broadcasting using two EDHCs, while the ratios appear in order 1.95, 1.92, 1.86, 1.78, and 1.68 for broadcasting using three EDHCs. Obviously, this ratio gets farther away from twice as the network dimension increases.

6 Concluding remarks

In this paper, we study the construction of multiple edge-disjoint Hamiltonian cycles as broadband channels in the crossed cube networks for fault-tolerant data broadcasting. The main contributions of this research are as follows:

(1) Using the result of [17] and the technique of edge exchange, we provide the construction of three EDHCs in CQ_6.
(2) By induction, we present a recursive algorithm to construct three EDHCs in CQ_n for $n \geqslant 7$.
(3) Based on (1) and (2), we simulate data broadcasting through the developed three EDHCs (resp. two EDHCs generated by Hung [18]) in CQ_n as the transmission channels, where $n \in \{6, 7, 8, 9, 10\}$.
(4) The comparison results show that broadcasting using three EDHCs is better than two EDHCs in performance, including the ASR in edge fault-tolerant broadcasting and two measures ABL and MBL related to broadcasting latency.

Before closing this paper, we will reflect on the research limitations. Although it is easy to implement a Hamiltonian path or cycle as a channel for data broadcasting, it is well known that such broadcasting delivery time is notoriously inefficient. Hence, if the number of nodes in the network is exponential, the topological scheme of broadcasting channels needs to be adequately adjusted. Usually, a tree structure can reduce the broadcasting delivery time by a logarithmic level, e.g., see [45–47].

In general, the reliability of a network depends on its connectivity, edge connectivity, or other related variants, which are important measures for the fault tolerance of networks. However, if the network topology is determined, these connectivities are invariants. Usually, a network with higher edge connectivity can construct more EDHCs. As future works, we are interested to see if there exist four EDHCs of CQ_n for $n \geqslant 8$. So far, this is still an open problem. Also, we can take experiments to simulate edge fault-tolerant data broadcasting under the conditional link faults, i.e., the condition means that every node must be incident with at least two fault-free edges [27, 28].

Acknowledgements This research was partially supported by MOST grants 111-2221-E-131-012 (K.-J. Pai), 111-2221-E-262-004 (R.-Y. Wu), 111-2221-E-141-007 (S.-L. Peng), and 111-2221-E-141-006 (J.-M. Chang) from the Ministry of Science and Technology, Taiwan.

Data availability All data generated or analyzed during this study are included in this published article.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest.

References

1. Lee S, Shin K (1994) Interleaved all-to-all reliable broadcast on meshes and hypercubes. IEEE Trans Parallel Distrib Syst 5(5):449–458
2. Leighton FT (1992) Introduction to parallel algorithms and architecture: arrays, trees, Hypercubes. Morgan Kaufmann, San Mateo, Calif
3. Lin T-J, Hsieh S-Y, Juan JS-T (2012) Embedding cycles and paths in product networks and their applications to multiprocessor systems. IEEE Trans Parallel Distrib Syst 23(6):1081–1089
4. Wang N-C, Yen C-P, Chu C-P (2005) Multicast communication in wormhole-routed symmetric networks with Hamiltonian cycle model. J Syst Arch 51(3):165–183
5. Bae MM, Bose B (2003) Edge disjoint Hamiltonian cycles in k-ary n-cubes and hypercubes. IEEE Trans Comput 52(10):1271–1284
6. Rowley R, Bose B (1991) Edge disjoint Hamiltonian cycles in de Bruijn networks. In: Proceedings of the 6th Distributed Memory Computing Conference, Portland, USA, pp 707–709
7. Xu J-M, Ma M (2009) A survey on cycle and path embedding in some networks. Front Math China 4(2):217–252
8. Garey MR, Johnson DS (1979) Computers and intractability: a guide to the theory of NP-completeness. Freeman, San Francisco
9. Barth D, Raspaud A (1994) Two edge-disjoint Hamiltonian cycles in the butterfly graph. Inf Process Lett 51(4):175–179
10. Albader B, Bose B (2016) Edge disjoint Hamiltonian cycles in Gaussian networks. IEEE Trans Comput 65(1):315–321
11. Hung R-W (2011) Embedding two edge-disjoint Hamiltonian cycles into locally twisted cubes. Theor Comput Sci 412:4747–4753
12. Li S-Y, Chang J-M, Pai K-J (2020) A parallel algorithm for constructing two edge-disjoint hamiltonian cycles in locally twisted cubes. In: Proceedings of International Computer Conference (ICS 2020), Tainan, Taiwan, 17–19 Dec, pp 116–119
13. Lü H, Wu T (2019) Edge-disjoint Hamiltonian cycles of balanced hypercubes. Inf Process Lett 144:25–30
14. Hung R-W (2012) Constructing two edge-disjoint Hamiltonian cycles and two-equal path cover in augmented cubes. IAENG Int'l J Comput Sci 39(1):42–49
15. Hung R-W, Chan S-J, Liao C-C (2012) Embedding two edge disjoint Hamiltonian cycles and two equal node disjoint cycles into twisted cubes. In: Proceedings of International Multi-Conference Engineers and Computer Scientists (IMECS 2012), Hong Kong, 14–6 Mar, pp 362–367
16. Wang Y, Fan J, Liu W, Wang X (2013) Embedding two edge-disjoint Hamiltonian cycles into parity cubes. Appl Mech Mater 336–338:2248–2251
17. Pai K-J (2020) A parallel algorithm for constructing two edge-disjoint Hamiltonian cycles in crossed cubes. In: Proceedings of 14th International Conference on Algorithmic Applications in Management (AAIM 2020). Lecture Notes in Computer Science, 12290: 448–455
18. Hung R-W (2015) The property of edge disjoint Hamiltonian cycles in transposition networks and hypercube like networks. Discrete Appl Math 181:109–122
19. Hussain ZA, Bose B, Al-Dhelaan A (2015) Edge disjoint Hamiltonian cycles in Eisenstein-Jacobi networks. J Parallel Distrib Comput 86:62–70
20. Petrovic V, Thomassen C (2006) Edge disjoint Hamiltonian cycles in hypertournaments. J Graph Theory 51(1):49–52
21. Huang C-H, Fang J-F, Yang C-Y (2004) Edge-disjoint Hamiltonian cycles of WK-recursive networks. In: Proceedings of 7th International Workshop on Applied Parallel Computing (PARA 2004). Lecture Notes in Computer Science, 3732: 1009–1104
22. Hussain ZA, Bose B, Al-Dhelaan A (2014) Generalized hypercubes: edge-disjoint Hamiltonian cycles and Gray codes. IEEE Trans Comput 63(2):375–382

23. Chen H-C, Kung T-L, Zou Y-H, Mao H-W (2015) The fault-tolerant Hamiltonian problems of crossed cube with path faults. IEICE Trans Infom Syst E98–D(12):2116–2122
24. Huang W-T, Chuang Y-C, Tan JJ-M, Hsu L-H (2002) On the fault-tolerant Hamiltonicity of faulty crossed cubes. IEICE Trans Fundam Electron Commun Comput Sci E85–A(6):1359–1370
25. Wang D (2008) On embedding hamiltonian cycles in crossed cubes. IEEE Trans Parallel Distrib Syst 19(3):334–346
26. Xu X, Zhang H, Wang Z, Zhang Q, Zhang P (2021) $(n-2)$-fault-tolerant edge-pancyclicity of crossed cubes CQ_n. Int J Found Comput Sci 32(3):289–304
27. Fu J-S, Hung H-S, Chen G-H (2009) Embedding fault-free cycles in crossed cubes with conditional link faults. J Supercomput 49(2):219–233
28. Hung H-S, Fu J-S, Chen G-H (2007) Fault-free Hamiltonian cycles in crossed cubes with conditional link faults. Inf Sci 177:5664–5674
29. Chen H-C (2018) The panpositionable panconnectedness of crossed cubes. J Supercomput 74(6):2638–2655
30. Chen H-C, Kung T-L, Hsu L-H (2011) Embedding a Hamiltonian cycle in the crossed cube with two required vertices in the fixed positions. Appl Math Comput 217(24):10058–10065
31. Chang J-M, Wang J-D, Yang J-S, Pai K-J (2014) A comment on "independent spanning trees in crossed cubes''. Inf Process Lett 114(12):734–739
32. Cheng B, Fan J, Jia X, Wang J (2013) Dimension-adjacent trees and parallel construction of independent spanning trees on crossed cubes. J Parallel Distrib Comput 73(5):641–652
33. Cheng B, Fan J, Jia X, Zhang S (2013) Independent spanning trees in crossed cubes. Inf Sci 233:276–289
34. Cheng B, Fan J, Lyu Q, Zhou J, Liu Z (2018) Constructing independent spanning trees with height n on the n-dimensional crossed cube. Future Gener Comput Syst 87:404–415
35. Cheng B, Wang D, Fan J (2017) Constructing completely independent spanning trees in crossed cubes. Discrete Appl Math 219:100–109
36. Pai K-J, Chang J-M (2016) Constructing two completely independent spanning trees in hypercube-variant networks. Theor Comput Sci 652:28–37
37. Pai K-J, Chang R-S, Wu R-Y, Chang J-M (2019) A two-stages tree-searching algorithm for finding three completely independent spanning trees. Theor Comput Sci 784:65–74
38. Pai K-J, Chang R-S, Wu R-Y, Chang J-M (2020) Three completely independent spanning trees of crossed cubes with application to secure-protection routing. Inf Sci 541:516–530
39. Wang X, Fan J, Zhang S, Yu J (2022) Node-to-set disjoint paths problem in cross-cubes. J Supercomput 78(1):1356–1380
40. Fan J (2002) Diagnosability of crossed cubes under the comparison diagnosis model. IEEE Trans Parallel Distrib Syst 13(10):1099–1104
41. Guo J, Li D, Lu M (2019) The g-good-neighbor conditional diagnosability of the crossed cubes under the PMC and MM* model. Theor Comput Sci 755:81–88
42. Chang C-P, Sung T-Y, Hsu L-H (2000) Edge congestion and topological properties of crossed cubes. IEEE Trans Parallel Distrib Syst 11(1):64–80
43. Efe K, Blackwell PK, Slough W, Shiau T (1994) Topological properties of the crossed cube architecture. Parallel Comput 20(12):1763–1775
44. Efe K (1992) The crossed cube architecture for parallel computation. IEEE Trans Parallel Distrib Syst 3(5):513–524
45. Fragopoulou P, Akl SG (1996) Edge-disjoint spanning trees on the star network with applications to fault tolerance. IEEE Trans Comput 45(2):174–185
46. Chen Y-S, Juang T-Y, Tseng E-H (2000) Efficient broadcasting in an arrangement graph using multiple spanning trees. IEICE Trans Fundam Electron Commun Comput Sci E83–A(1):139–149
47. Yang J-S, Chan H-C, Chang J-M (2011) Broadcasting secure messages via optimal independent spanning trees in folded hypercubes. Discrete Appl Math 159:1254–1263

Authors and Affiliations

Kung-Jui Pai[1] · Ro-Yu Wu[2] · Sheng-Lung Peng[3] · Jou-Ming Chang[4]

Kung-Jui Pai
poter@mail.mcut.edu.tw

Ro-Yu Wu
eric@mail.lhu.edu.tw

Sheng-Lung Peng
slpeng@ntub.edu.tw

[1] Department of Industrial Engineering and Management, Ming Chi University of Technology, New Taipei City 243303, Taiwan

[2] Department of Industrial Management, Lunghwa University of Science and Technology, Taoyüan 333326, Taiwan

[3] Department of Product Innovation and Entrepreneurship, National Taipei University of Business, Taoyüan 324022, Taiwan

[4] Institute of Information and Decision Sciences, National Taipei University of Business, Taipei 10051, Taiwan