IEICE TRANSACTIONS

on Information and Systems

DOI:10.1587/transinf.2024EDP7040

Publicized:2024/05/17

This advance publication article will be replaced by the finalized version after proofreading.

A PUBLICATION OF THE INFORMATION AND SYSTEMS SOCIETY
The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER

Node-to-node and Node-to-set Disjoint Paths Problems in Bicubes

Arata KANEKO[†], **Htoo Htoo Sandi KYAW**[†], ***Nonmembers***, **Kunihiro FUJIYOSHI**[†], ***and*** **Keiichi KANEKO**[†a)], ***Members***

SUMMARY In this paper, we propose two algorithms, B-N2N and B-N2S, that solve the node-to-node and node-to-set disjoint paths problems in the bicube, respectively. We prove their correctness and that the time complexities of the B-N2N and B-N2S algorithms are $O(n^2)$ and $O(n^2 \log n)$, respectively, if they are applied in an n-dimensional bicube with $n \geq 5$. Also, we prove that the maximum lengths of the paths generated by B-N2N and B-N2S are both $n+2$. Furthermore, we have shown that the algorithms can be applied in the locally twisted cube, too, with the same performance.
key words: *bicube, hypercube, interconnection network, locally twisted cube, massively parallel system, topology*

1. Introduction

The hypercube [1] was once a very popular topology for interconnection networks of massively parallel systems, and it has many variants. The bicube [2] is such a topology and it attracts much attention [3]–[7] because it can interconnect the same number of nodes with the same degree as the hypercube while its diameter is almost half of that of the hypercube. In addition, the bicube preserves the property of node symmetry.

In this paper, we propose two algorithms, B-N2N and B-N2S, that solve the node-to-node and node-to-set disjoint paths problems in the bicube, respectively. There is a generic algorithm [8] that solves the problems in cube-based topologies. If we apply it to the problems in an n-dimensional bicube ($n \geq 3$), we can generate n node-disjoint paths whose lengths are at most $2n-1$ in $O(n^4)$ time for both problems. On the other hand, B-N2N generates n node-disjoint paths of lengths at most $n+2$ in $O(n^2)$ time while B-N2S generates n node-disjoint paths of lengths at most $n+2$ in $O(n^2 \log n)$ time. B-N2N and B-N2S use the algorithm proposed by Bossard and Kaneko [9], which we call H-N2S, because a bicube consists of two hypercubes with bijective or one-to-one edges between them. H-N2S solves the node-to-set disjoint paths problem in the hypercube. Our algorithms, B-N2N and B-N2S, can be applied in the locally twisted cube with the same performance because a locally twisted cube also consists of two hypercubes with bijective edges between them [10].

[†]The authors are with Graduate School of Engineering, Tokyo University of Agriculture and Technology, Koganei-shi, 184-8588 Japan.

a) E-mail: k1kaneko@cc.tuat.ac.jp (Corresponding author).

Given a source node $\boldsymbol{s}$ and a destination node $\boldsymbol{d}$ in a k-connected graph, the node-to-node disjoint paths problem is to generate k paths U_i: $\boldsymbol{s} \rightsquigarrow \boldsymbol{d}$ $(1 \leq i \leq k)$ such that U_i $(1 \leq i \leq k)$ are node-disjoint except for $\boldsymbol{s}$ and $\boldsymbol{d}$. In addition, given a source node $\boldsymbol{s}$ and a set of k destination nodes $\{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_k\}$ in a k-connected graph, the node-to-set disjoint paths problem is to generate k paths U_i: $\boldsymbol{s} \rightsquigarrow \boldsymbol{d}_i$ $(1 \leq i \leq k)$ such that U_i $(1 \leq i \leq k)$ are node-disjoint except for $\boldsymbol{s}$. The node-to-node disjoint paths problem [11]–[16] and the node-to-set disjoint paths problem [8], [9], [17]–[23] are important issues in parallel and distributed computation as well as the set-to-set disjoint paths problem [18], [24]–[28]: given a set of k source nodes $\{\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_k\}$ and a set of k destination nodes $\{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_k\}$ in a k-connected graph, the set-to-set disjoint paths problem is to generate k paths U_i: $\boldsymbol{s}_i \rightsquigarrow \boldsymbol{d}_{j_i}$ $(1 \leq i \leq k, \{j_1, j_2, \ldots, j_k\} = \{1, 2, \ldots, k\})$ such that U_i $(1 \leq i \leq k)$ are node-disjoint. Generating disjoint paths in a massively parallel system has many applications. For example, multiple pairs of nodes can establish the full-bandwidth communication over a network simultaneously by using the circuit switching. The circuit switching provides an optimal data transfer performance because it does not require any switching inside the routers of intermediate nodes. Also, the circuit switching does not allow any interference with other communications, ensuring security and privacy. The studies of the node-disjoint paths problems with respect to some cube-based topologies are summarized in **Table 1**.

In the rest of this paper, we use 'disjoint' instead of 'node-disjoint' for simplicity.

2. Preliminaries

In this section, we give the definitions of related topics and the properties of the bicube. Generally, we adopt the notations and terminology from the traditional graph theory. For example, a path in a graph $G(V, E)$ is an alternate sequence of nodes and edges: $\boldsymbol{u}_1, (\boldsymbol{u}_1, \boldsymbol{u}_2), \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{l-1}, (\boldsymbol{u}_{l-1}, \boldsymbol{u}_l), \boldsymbol{u}_l$ for $\boldsymbol{u}_i \in V$ $(1 \leq i \leq l)$, and we use a shorthand $\boldsymbol{u}_1 \rightarrow \boldsymbol{u}_2 \rightarrow \cdots \rightarrow \boldsymbol{u}_l$ or $\boldsymbol{u}_1 \rightsquigarrow \boldsymbol{u}_l$ if the intermediate nodes are not important. The length of a path is the number of edges included in the path. Let us consider two paths P: $\boldsymbol{u} \rightsquigarrow \boldsymbol{v}$ and Q: $\boldsymbol{x} \rightsquigarrow \boldsymbol{y}$. Then, if P and Q do not have any common node, they are disjoint. If P and Q do not have any common node except for $\boldsymbol{u}(=\boldsymbol{x})$, they are disjoint except for $\boldsymbol{u}(=\boldsymbol{x})$. If P and Q do not have any common node except for $\boldsymbol{u}(=\boldsymbol{x})$ and $\boldsymbol{v}(=\boldsymbol{y})$, they are

Table 1 Time complexities and maximum path lengths of node-disjoint paths routing algorithms for constructing n disjoint paths in n-dimensional cube-based topologies.

topology	diameter	node-to-node		node-to-set		set-to-set	
		time	length	time	length	time	length
Hypercube	n	$O(n^2)$ [11]	$n+1$ [11]	$O(n^2)$ [9]	$n+1$ [9]	$O(n^2 \log n)$ [18]	$2n$ [18]
Bicube	$\lceil (n+1)/2 \rceil^\dagger$	—	—	—	—	—	—
Locally Twisted Cube	$\lceil (n+3)/2 \rceil^\ddagger$	—	—	—	—	—	—
Twisted Cube	$\lceil (n+1)/2 \rceil$	—	$\lceil n/2 \rceil + 2^\S$ [13]	—	—	—	—
Crossed Cube	$\lceil (n+1)/2 \rceil$	$O(n^2)$ [12]	$3n-5$ [12]	—	—	—	—
Twisted Crossed Cube	$\lceil (n+1)/2 \rceil$	$O(n^2)$ [16]	$4n-8$ [16]	—	—	—	—
0-Möbius Cube	$\lceil (n+2)/2 \rceil$	$O(n^2)$ [15]	$3n-5$ [15]	$O(n^4)$ [22]	$2n-1$ [22]	$O(n^6)$ [27]	$2n-2$ [27]
1-Möbius Cube	$\lceil (n+1)/2 \rceil$	$O(n^2)$ [15]	$3n-5$ [15]	$O(n^4)$ [22]	$2n-1$ [22]	$O(n^6)$ [27]	$2n-2$ [27]

$\dagger$: $n \geq 7$, $\ddagger$: $n \geq 5$, $\S$: n is odd.

disjoint except for $\boldsymbol{u}(=\boldsymbol{x})$ and $\boldsymbol{v}(=\boldsymbol{y})$.

Definition 1: An n-dimensional hypercube, H_n, is an undirected graph whose node set is $\{0,1\}^n$. Given two nodes $\boldsymbol{u}$ and $\boldsymbol{v}$ in H_n, $\boldsymbol{u}$ and $\boldsymbol{v}$ are neighboring if and only if $h(\boldsymbol{u},\boldsymbol{v}) = 1$ where $h(\boldsymbol{u},\boldsymbol{v})$ represents the Hamming distance between $\boldsymbol{u}$ and $\boldsymbol{v}$. □

The number of nodes and the diameter of H_n are 2^n and n, respectively. H_n is symmetric and its degree is n. H_n has a recursive structure such that it consists of two H_{n-1}'s. Also, H_n has a shortest-path routing algorithm SPR that generates one of the shortest paths between any pair of nodes whose length is at most n in $O(n)$ time.

Definition 2: Given a bit sequence $\boldsymbol{u} = (u_n, u_{n-1}, \ldots, u_1) (\in \{0,1\}^n)$, define a function $p(\boldsymbol{u})$ by $p(\boldsymbol{u}) = u_n \oplus u_{n-1} \oplus \cdots \oplus u_1$ where '$\oplus$' represents the exclusive-or operation: $0 \oplus 0 = 1 \oplus 1 = 0$ and $1 \oplus 0 = 0 \oplus 1 = 1$. Then, given a pair of bit sequences $\boldsymbol{u}, \boldsymbol{v} \in \{0,1\}^n$ with an even n, $\boldsymbol{u}$ and $\boldsymbol{v}$ are in lp-relation if and only if '$\boldsymbol{u} = \boldsymbol{v}$ and $p(\boldsymbol{u}) = p(\boldsymbol{v}) = 0$' or '$\boldsymbol{u} = \overline{\boldsymbol{v}}$ and $p(\boldsymbol{u}) = p(\boldsymbol{v}) = 1$'. □

For instance, for two bit sequences $(0,1,1,0,0,1)$ and $(1,0,0,1,1,0)$, they are in lp-relation because $(0,1,1,0,0,1) = (\overline{1,0,0,1,1,0})$ and $p(0,1,1,0,0,1) = p(1,0,0,1,1,0) = 1$.

Definition 3: An n-dimensional bicube, B_n ($n \geq 3$), is an undirected graph whose node set is $\{0,1\}^n$. Given a node $\boldsymbol{u} = (u_n, u_{n-1}, \ldots, u_1)$ in B_n, it has n neighboring nodes $\boldsymbol{u}^{(i)}$ ($1 \leq i \leq n$) where $\boldsymbol{u}^{(i)}$ ($1 \leq i \leq n-1$) are given by $\boldsymbol{u}^{(i)} = (u_n, u_{n-1}, \ldots, u_{i+1}, \overline{u_i}, u_{i-1}, \ldots, u_1)$ and $\boldsymbol{u}^{(n)}$ is given depending on the parity of n. That is, if n is odd, $\boldsymbol{u}^{(n)} = (\overline{u_n}, v_{n-1}, v_{n-2}, \ldots, v_1)$, where $(v_{n-1}, v_{n-2}, \ldots, v_1)$ is the bit sequence that is in lp-relation with $(u_{n-1}, u_{n-2}, \ldots, u_1)$. If n is even, $\boldsymbol{u}^{(n)} = (\overline{u_n}, u_{n-1}, v_{n-2}, \ldots, v_1)$ where $(v_{n-2}, v_{n-3}, \ldots, v_1)$ is the bit sequence that is in lp-relation with $(u_{n-2}, u_{n-3}, \ldots, u_1)$. Note that $(\boldsymbol{u}^{(i)})^{(j)}$ ($1 \leq i, j \leq n$) is denoted by $\boldsymbol{u}^{(i,j)}$ in short. □

As an example, B_5 is shown in **Fig. 1**. B_n has the following properties [2]. The number of nodes and the diameter of B_n are 2^n and $\lceil (n+1)/2 \rceil$ ($n \geq 7$), respectively. The diameters of B_3, B_4, B_5, and B_6 are 3, 4, 4, and 5, respectively. B_n is a node-symmetric graph whose degree is n. Because the bicube is bipartite, it does not include any cycle with an odd length. Now, let B_n^0 and B_n^1 be the subgraphs induced by the node sets $\{(u_n, u_{n-1}, \ldots, u_1) \mid u_n = 0\} (\subset V(B_n))$ and $\{(u_n, u_{n-1}, \ldots, u_1) \mid u_n = 1\} (\subset V(B_n))$, respectively. Then, B_n^0 and B_n^1 are both isomorphic to an $(n-1)$-dimensional hypercube, H_{n-1}. In other words, B_n consists of two H_{n-1}'s. The left and right subgraphs, B_5^0 and B_5^1, form two distinct H_4's in Fig. 1, in which $(0,0,1,1,1) (\in B_5^0)$ is connected to $(1,1,0,0,0) (\in B_5^1)$ while it is connected to $(1,0,1,1,1)$ in a 5-dimensional hypercube, H_5, for example.

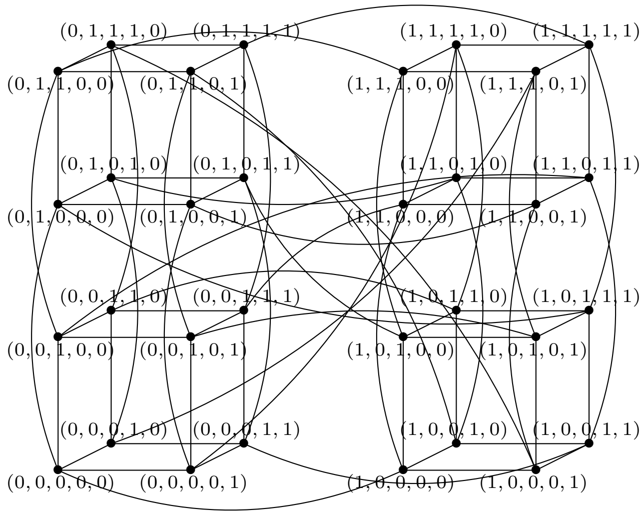

Fig. 1 Example of a 5-dimensional bicube, B_5.

Lemma 1: For a node $\boldsymbol{u} \in V(B_n^b)$ ($b \in \{0,1\}$), there are n paths from $\boldsymbol{u}$ such that the other terminal nodes are included in $V(B_n^{\bar{b}})$, the paths are disjoint except for $\boldsymbol{u}$, and their lengths are at most 2.
(Proof) We can generate n paths L_i ($1 \leq i \leq n$) as follows:

$$L_i : \begin{cases} \boldsymbol{u} \to \boldsymbol{u}^{(i)} \to \boldsymbol{u}^{(i,n)} & (1 \leq i \leq n-1), \\ \boldsymbol{u} \to \boldsymbol{u}^{(n)} & (i = n). \end{cases}$$

Then, the path lengths are at most 2. $\boldsymbol{u}^{(i,n)}$ ($1 \leq i \leq n-1$) and $\boldsymbol{u}^{(n)}$ are included in $V(B_n^{\bar{b}})$. Because $\boldsymbol{u}^{(i)}$ ($1 \leq i \leq$

$n-1$) and $\boldsymbol{u}^{(n)}$ are distinct neighboring nodes of $\boldsymbol{u}$, and B_n^b and $B_n^{\bar{b}}$ are connected by one-to-one edges, the paths are disjoint except for $\boldsymbol{u}$. □

3. B-N2N Algorithm

In this section, we describe our algorithm B-N2N that solves the node-to-node disjoint paths problem in an n-dimensional bicube, B_n. Let $\boldsymbol{s}$ and $\boldsymbol{d}$ be the source node and the destination node, respectively. For n with $3 \le n \le 4$, B_n is isomorphic to H_n. Hence, it is trivial to find n paths between $\boldsymbol{s}$ and $\boldsymbol{d}$ that are disjoint except for $\boldsymbol{s}$ and $\boldsymbol{d}$ in $O(n^2)$ time, whose lengths are at most $n+1$ by using the algorithm proposed by Saad and Schultz [11]. Thus, we assume that $n \ge 5$. Without loss of generality, we can assume that $\boldsymbol{s} \in V(B_n^0)$. Then B-N2N is divided into two cases depending on the position of the destination node.

3.1 B-N2N Case 1 ($\boldsymbol{d} \in V(B_n^0)$)

Step 1 Apply H-N2S in B_n^0 to generate $(n-1)$ paths P_i: $\boldsymbol{s} \rightsquigarrow \boldsymbol{d}^{(i)}$ $(1 \le i \le n-1)$ that are disjoint except for $\boldsymbol{s}$.
Step 2 Select $(n-1)$ edges $\boldsymbol{d}^{(i)} \to \boldsymbol{d}$ $(1 \le i \le n-1)$.
Step 3 Select two edges $\boldsymbol{s} \to \boldsymbol{s}^{(n)}$ and $\boldsymbol{d}^{(n)} \to \boldsymbol{d}$.
Step 4 Apply SPR in B_n^1 to generate the shortest path R: $\boldsymbol{s}^{(n)} \rightsquigarrow \boldsymbol{d}^{(n)}$ (**Fig. 2**).

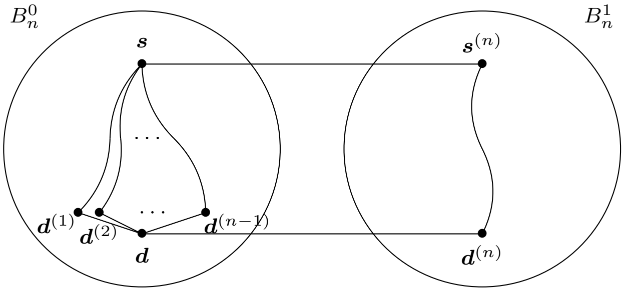

Fig. 2 After Step 4 of B-N2N Case 1

Step 5 Construct n paths, U_i $(1 \le i \le n)$, that are disjoint except for $\boldsymbol{s}$ and $\boldsymbol{d}$ as follows:

$$U_i : \begin{cases} \boldsymbol{s} \overset{P_i}{\rightsquigarrow} \boldsymbol{d}^{(i)} \to \boldsymbol{d} & (1 \le i \le n-1) \\ \boldsymbol{s} \to \boldsymbol{s}^{(i)} \overset{R}{\rightsquigarrow} \boldsymbol{d}^{(i)} \to \boldsymbol{d} & (i = n) \end{cases}$$

3.2 B-N2N Case 2 ($\boldsymbol{d} \in V(B_n^1)$)

Step 1 Select the edge $\boldsymbol{s} \to \boldsymbol{s}^{(n)}$.
Step 2 Apply SPR in B_n^1 to generate the shortest path R: $\boldsymbol{s}^{(n)} \rightsquigarrow \boldsymbol{d}$. Let $\boldsymbol{d}^{(l)}$ be the neighboring node of $\boldsymbol{d}$ that is included in R.
Step 3 Apply H-N2S in B_n^0 to generate $(n-1)$ paths, P_i $(1 \le i(\ne l) \le n)$, that are disjoint except for $\boldsymbol{s}$ as follows:

$$P_i : \begin{cases} \boldsymbol{s} \rightsquigarrow \boldsymbol{d}^{(i,n)} & (1 \le i(\ne l) \le n-1) \\ \boldsymbol{s} \rightsquigarrow \boldsymbol{d}^{(i)} & (i = n) \end{cases}$$

Step 4 Select $(n-2)$ paths Q_i: $\boldsymbol{d}^{(i,n)} \to \boldsymbol{d}^{(i)} \to \boldsymbol{d}$ $(1 \le i(\ne l) \le n-1)$ of length 2 and a path Q_n: $\boldsymbol{d}^{(n)} \to \boldsymbol{d}$ of length 1 (**Fig. 3**).

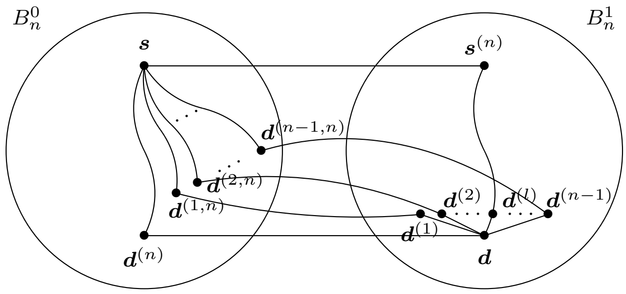

Fig. 3 After Step 4 of B-N2N Case 2

Step 5 Construct n paths, U_i $(1 \le i \le n)$, that are disjoint except for $\boldsymbol{s}$ and $\boldsymbol{d}$ as follows:

$$U_i : \begin{cases} \boldsymbol{s} \overset{P_i}{\rightsquigarrow} \boldsymbol{d}^{(i,n)} \overset{Q_i}{\to} \boldsymbol{d}^{(i)} \overset{Q_i}{\to} \boldsymbol{d} & (1 \le i(\ne l) \le n-1) \\ \boldsymbol{s} \to \boldsymbol{s}^{(n)} \overset{R}{\rightsquigarrow} \boldsymbol{d}^{(i)} \overset{R}{\to} \boldsymbol{d} & (i = l) \\ \boldsymbol{s} \overset{P_i}{\rightsquigarrow} \boldsymbol{d}^{(i)} \overset{Q_i}{\to} \boldsymbol{d} & (i = n) \end{cases}$$

4. B-N2S Algorithm

In this section, we describe our algorithm B-N2S that solves the node-to-set disjoint paths problem in an n-dimensional bicube, B_n. Let $\boldsymbol{s}$ be the source node and $\{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_n\}$ be the set of n destination nodes. For n with $3 \le n \le 4$, B_n is isomorphic to H_n. Hence, it is trivial to find n paths $\boldsymbol{s} \rightsquigarrow \boldsymbol{d}_i$ $(1 \le i \le n)$ that are disjoint except for $\boldsymbol{s}$ in $O(n^2)$ time, whose lengths are at most $n+1$ by using the algorithm proposed by Bossard and Kaneko [9]. Thus, we assume that $n \ge 5$. Without loss of generality, we can assume that $\boldsymbol{s} \in V(B_n^0)$ and $\{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_n\} \cap V(B_n^0) = \{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_l\}$. If $\{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_n\} \cap V(B_n^0) = \emptyset$, we execute the steps as $l = 0$. Then B-N2S is divided into two cases depending on the distribution of the destination nodes.

4.1 B-N2S Case 1 ($l = n$)

Step 1 Apply H-N2S in B_n^0 to generate $(n-1)$ paths P_i: $\boldsymbol{s} \rightsquigarrow \boldsymbol{d}_i$ $(1 \le i \le n-1)$ that are disjoint except for $\boldsymbol{s}$.
Step 2 If $\boldsymbol{d}_n$ is included in one of the paths generated in Step 1, say P_x: $\boldsymbol{s} \rightsquigarrow \boldsymbol{d}_x$, let P_x: $\boldsymbol{s} \rightsquigarrow \boldsymbol{d}_n$ by discarding the subpath $\boldsymbol{d}_n \rightsquigarrow \boldsymbol{d}_x$, and exchange the indices of $\boldsymbol{d}_x$ and $\boldsymbol{d}_n$.
Step 3 Select two edges $\boldsymbol{s} \to \boldsymbol{s}^{(n)}$ and $\boldsymbol{d}_n^{(n)} \to \boldsymbol{d}_n$.
Step 4 Apply SPR in B_n^1 to generate the shortest path R: $\boldsymbol{s}^{(n)} \rightsquigarrow \boldsymbol{d}_n^{(n)}$ (**Fig. 4**).
Step 5 Construct n paths, U_i $(1 \le i \le n)$, that are disjoint

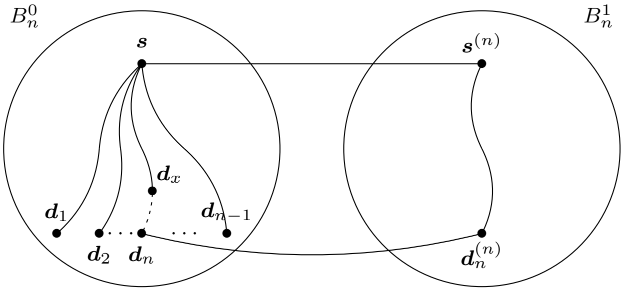

Fig. 4 After Step 4 of B-N2S Case 1

except for $\boldsymbol{s}$ as follows:

$$U_i : \begin{cases} \boldsymbol{s} \overset{P_i}{\rightsquigarrow} \boldsymbol{d}_i & (1 \leq i \leq n-1) \\ \boldsymbol{s} \to \boldsymbol{s}^{(n)} \overset{R}{\rightsquigarrow} \boldsymbol{d}_i^{(n)} \to \boldsymbol{d}_i & (i = n) \end{cases}$$

4.2 B-N2S Case 2 ($l < n$)

Step 1 For each node of $\boldsymbol{d}_i$ ($l+1 \leq i \leq n$), find a path Q_i: $\boldsymbol{d}'_i (\in V(B_n^0)) \rightsquigarrow \boldsymbol{d}_i$ of lengths at most 2 such that it is disjoint from other paths Q_j ($l+1 \leq j (\neq i) \leq n$) and does not include destination nodes $\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_l$.
Step 2 Select the edge $\boldsymbol{s} \to \boldsymbol{s}^{(n)}$.
Step 3 Apply SPR in B_n^1 to generate the shortest path R: $\boldsymbol{s}^{(n)} \rightsquigarrow \boldsymbol{d}_n$.
Step 4 If the path R does not include any node on the paths generated in Step 1, go to Step 5. Otherwise, let $\hat{\boldsymbol{d}}_x$ be the closest one to $\boldsymbol{s}^{(n)}$ along R. Also, let Q_x: $\boldsymbol{d}'_x \rightsquigarrow \hat{\boldsymbol{d}}_x \rightsquigarrow \boldsymbol{d}_x$ be the path to which $\hat{\boldsymbol{d}}_x$ belongs (**Fig. 5**). Then, discard the subpath $\hat{\boldsymbol{d}}_x \rightsquigarrow \boldsymbol{d}_n$ of R,

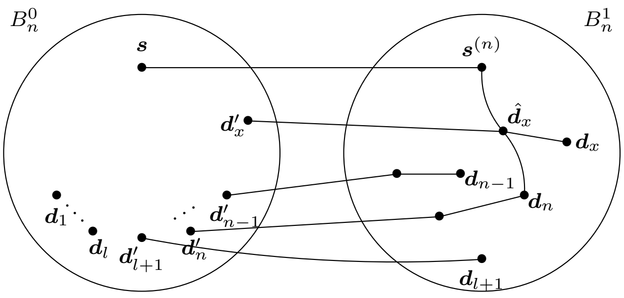

Fig. 5 During Step 4 of B-N2S Case 2

and select R: $\boldsymbol{s}^{(n)} \rightsquigarrow \hat{\boldsymbol{d}}_x \rightsquigarrow \boldsymbol{d}_x$. Also, exchange the indices of Q_x and Q_n, the indices of $\boldsymbol{d}_x$ and $\boldsymbol{d}_n$, and the indices of $\boldsymbol{d}'_x$ and $\boldsymbol{d}'_n$.
Step 5 Discard the path Q_n (**Fig. 6**).
Step 6 Apply H-N2S in B_n^0 to generate $(n-1)$ paths P_i: $\boldsymbol{s} \rightsquigarrow \boldsymbol{d}_i$ ($1 \leq i \leq l$) and P_i: $\boldsymbol{s} \rightsquigarrow \boldsymbol{d}'_i$ ($l+1 \leq i \leq n-1$) that are disjoint except for $\boldsymbol{s}$.
Step 7 Construct n paths, U_i ($1 \leq i \leq n$), that are disjoint except for $\boldsymbol{s}$ as follows:

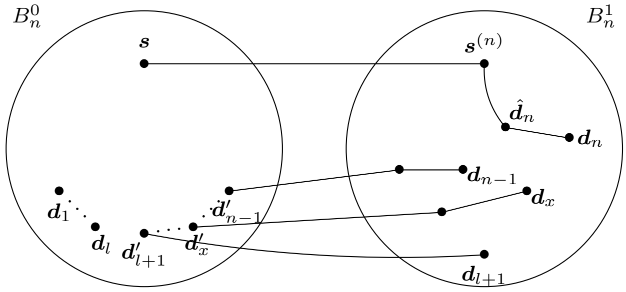

Fig. 6 After Step 5 of B-N2S Case 2

$$U_i : \begin{cases} \boldsymbol{s} \overset{P_i}{\rightsquigarrow} \boldsymbol{d}_i & (1 \leq i \leq l) \\ \boldsymbol{s} \overset{P_i}{\rightsquigarrow} \boldsymbol{d}'_i \overset{Q_i}{\rightsquigarrow} \boldsymbol{d}_i & (l+1 \leq i \leq n-1) \\ \boldsymbol{s} \to \boldsymbol{s}^{(n)} \overset{R}{\rightsquigarrow} \boldsymbol{d}_i & (i = n) \end{cases}$$

5. Correctness and Complexities

Lemma 2: For a source node $\boldsymbol{s}$ and a set of n destination nodes $\{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_n\}$ in an n-dimensional hypercube, the H-N2S algorithm by Bossard and Kaneko [9] generates n paths from $\boldsymbol{s}$ to $\boldsymbol{d}_i$ ($1 \leq i \leq n$) of lengths at most $n+1$ that are disjoint except for $\boldsymbol{s}$ in $O(n^2)$ time.
(Proof) From [9]. □

Lemma 3: In Case 1, for a source node $\boldsymbol{s}$ and a destination node $\boldsymbol{d}$ in an n-dimensional bicube with $n \geq 5$, B-N2N takes $O(n^2)$ time to generate n paths from $\boldsymbol{s}$ to $\boldsymbol{d}$ of lengths at most $n+1$ that are disjoint except for $\boldsymbol{s}$ and $\boldsymbol{d}$.
(Proof) In Step 1, H-N2S takes $O(n^2)$ time to generate $(n-1)$ paths P_i: $\boldsymbol{s} \rightsquigarrow \boldsymbol{d}^{(i)}$ ($1 \leq i \leq n-1$) of lengths at most n that are disjoint except for $\boldsymbol{s}$ from Lemma 2. In Step 2, it takes $O(n)$ time to select $(n-1)$ edges. In Step 3, it takes $O(1)$ time to select two edges. In Step 4, SPR takes $O(n)$ time to generate the shortest path whose length is at most $n-1$. The total time complexity of B-N2N in Case 1 is $O(n^2)$, and the maximum path length is $n+1$. The paths U_i: $\boldsymbol{s} \overset{P_i}{\rightsquigarrow} \boldsymbol{d}^{(i)} \to \boldsymbol{d}$ ($1 \leq i \leq n-1$) are disjoint except for $\boldsymbol{s}$ and $\boldsymbol{d}$ because P_i ($1 \leq i \leq n-1$) are generated by H-N2S. The path U_n is disjoint from other paths U_i ($1 \leq i \leq n-1$) except for $\boldsymbol{s}$ and $\boldsymbol{d}$ because it is outside of B_n^0 other than $\boldsymbol{s}$ and $\boldsymbol{d}$. □

Lemma 4: In Case 2, for a source node $\boldsymbol{s}$ and a destination node $\boldsymbol{d}$ in an n-dimensional bicube with $n \geq 5$, B-N2N takes $O(n^2)$ time to generate n paths from $\boldsymbol{s}$ to $\boldsymbol{d}$ of lengths at most $n+2$ that are disjoint except for $\boldsymbol{s}$ and $\boldsymbol{d}$.
(Proof) In Step 1, it takes $O(1)$ time to select the edge. In Step 2, SPR takes $O(n)$ time to generate the shortest path R of length at most $n-1$. It takes $O(1)$ time to find l of $\boldsymbol{d}^{(l)}$. In Step 3, H-N2S takes $O(n^2)$ time to generate $(n-1)$ paths P_i ($1 \leq i (\neq l) \leq n$) of lengths at most n that are disjoint except for $\boldsymbol{s}$ from Lemma 2. In Step 4, it takes $O(n)$ time to generate Q_i ($1 \leq i (\neq l) \leq n$) of lengths at most 2. The total time complexity of B-N2N in Case 2 is $O(n^2)$, and the maximum path length is $n+2$. U_l is disjoint from other U_i

$(1 \le i(\neq l) \le n)$ except for $\boldsymbol{s}$ and $\boldsymbol{d}$ because it is outside of B_n^0 other than $\boldsymbol{s}$ and it does not include any neighboring node of $\boldsymbol{d}$ except for $\boldsymbol{d}^{(l)}$. U_i $(1 \le i(\neq l) \le n)$ are disjoint except for $\boldsymbol{s}$ and $\boldsymbol{d}$ among others because P_i $(1 \le i(\neq l) \le n)$ are generated by H-N2S and Q_i $(1 \le i(\neq l) \le n)$ include distinct nodes $\boldsymbol{d}^{(i,n)}$ $(1 \le i(\neq l) \le n-1)$ and $\boldsymbol{d}^{(i)}$ $(1 \le i(\neq l) \le n)$. Consequently, U_i $(1 \le i \le n)$ are disjoint except for $\boldsymbol{s}$ and $\boldsymbol{d}$. □

Theorem 1: For a source node $\boldsymbol{s}$ and a destination node $\boldsymbol{d}$ in an n-dimensional bicube with $n \ge 5$, the B-N2N algorithm takes $O(n^2)$ time and it generates n paths from $\boldsymbol{s}$ to $\boldsymbol{d}$ of lengths at most $n+2$ that are disjoint except for $\boldsymbol{s}$ and $\boldsymbol{d}$.
(Proof) From Lemmas 3 and 4. □

Lemma 5: In Case 1, for a source node $\boldsymbol{s}$ and a set of n destination nodes $\{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_n\}$ in an n-dimensional bicube with $n \ge 5$, B-N2S takes $O(n^2)$ time to generate n paths from $\boldsymbol{s}$ to $\boldsymbol{d}_i$ $(1 \le i \le n)$ of lengths at most $n+1$ that are disjoint except for $\boldsymbol{s}$.
(Proof) In Step 1, H-N2S takes $O(n^2)$ time to generate $(n-1)$ paths P_i: $\boldsymbol{s} \rightsquigarrow \boldsymbol{d}_i$ $(1 \le i \le n-1)$ of lengths at most n that are disjoint except for $\boldsymbol{s}$ from Lemma 2. In Step 2, it takes $O(n^2)$ time to check if $\boldsymbol{d}_n$ is included in one of the paths generated in Step 1. It takes $O(n)$ time to discard the subpath and exchange the indices of $\boldsymbol{d}_x$ and $\boldsymbol{d}_n$. In Step 3, it takes $O(1)$ time to select two edges. In Step 4, SPR takes $O(n)$ time to generate the shortest path whose length is at most $n-1$. The total time complexity of B-N2S in Case 1 is $O(n^2)$, and the maximum path length is $n+1$. The paths U_i $(1 \le i \le n-1)$ are disjoint except for $\boldsymbol{s}$ because they are generated by H-N2S. The path U_n is disjoint from other paths U_i $(1 \le i \le n-1)$ except for $\boldsymbol{s}$ because it is outside of B_n^0 other than $\boldsymbol{s}$ and $\boldsymbol{d}_n$. □

Lemma 6: In Step 1 of Case 2, B-N2S can find the path Q_i for each destination node $\boldsymbol{d}_i$ $(l+1 \le i \le n)$.
(Proof) From Lemma 1, there are n candidate paths for Q_i. Destination nodes $\boldsymbol{d}_j$ $(1 \le j \le l$ or $i+1 \le j \le n)$ can block at most one of them. Also, each of the paths Q_j: $\boldsymbol{d}'_j \rightsquigarrow \boldsymbol{d}_j$ $(l+1 \le j \le i-1)$ can block at most one of the candidate paths because there is no cycle of length 3 in B_n. Note that because B_n^0 and B_n^1 are connected by bijective edges, the edge $\boldsymbol{d}_j^{(n)} \to \boldsymbol{d}_j$ of Q_j cannot block two candidate paths at a time. Hence, B-N2S can find at least one of n candidates for Q_i. □

Lemma 7: In Case 2, for a source node $\boldsymbol{s}$ and a set of n destination nodes $\{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_n\}$ in an n-dimensional bicube with $n \ge 5$, B-N2S takes $O(n^2 \log n)$ time to generate n paths from $\boldsymbol{s}$ to $\boldsymbol{d}_i$ $(1 \le i \le n)$ of lengths at most $n+2$ that are disjoint except for $\boldsymbol{s}$.
(Proof) In Step 1, it takes $O(n^2 \log n)$ time to find Q_i $(l+1 \le i \le n)$ of lengths at most 2 by using a balanced binary search tree. In Step 2, it takes $O(1)$ time to select the edge. In Step 3, SPR takes $O(n)$ time to generate R of length at most $n-1$. In Step 4, it takes $O(n^2)$ time to check if R includes a node on the paths generated in Step 1. It takes $O(n)$ time to find $\hat{\boldsymbol{d}}_x$ and discard the subpath, update R, and exchange the indices. In Step 5, it takes $O(1)$ to discard Q_n. In Step 6, H-N2S takes $O(n^2)$ time to generate P_i $(1 \le i \le n-1)$ of lengths at most n from Lemma 2. The total time complexity of B-N2S in Case 2 is $O(n^2 \log n)$, and the maximum path length is $n+2$. U_i $(1 \le i \le l)$ are disjoint among others except for $\boldsymbol{s}$ because they are generated by H-N2S. Also, they are disjoint from U_j $(l+1 \le j \le n-1)$ except for $\boldsymbol{s}$ because P_j $(l+1 \le j \le n-1)$ are generated by H-N2S and Q_j $(l+1 \le j \le n-1)$ are outside of B_n^0 other than $\boldsymbol{d}'_j$. Moreover, the paths are disjoint from U_n except for $\boldsymbol{s}$ because it is outside of B_n^0 other than $\boldsymbol{s}$. U_i $(l+1 \le i \le n-1)$ are disjoint among others except for $\boldsymbol{s}$ because P_i $(l+1 \le i \le n-1)$ are generated by H-N2S and Q_i $(l+1 \le i \le n-1)$ are generated in Step 1 such that they are disjoint. In addition, the paths are disjoint from U_n except for $\boldsymbol{s}$ because R is outside of B_n^0 other than $\boldsymbol{s}$, and it is generated such that it is ensured to be disjoint from Q_i $(l+1 \le i \le n-1)$ in Step 4. Consequently, U_i $(1 \le i \le n)$ are disjoint except for $\boldsymbol{s}$. □

Theorem 2: For a source node $\boldsymbol{s}$ and a set of n destination nodes $\{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_n\}$ in an n-dimensional bicube with $n \ge 5$, the B-N2S algorithm takes $O(n^2 \log n)$ time and it generates n paths from $\boldsymbol{s}$ to $\boldsymbol{d}_i$ $(1 \le i \le n)$ of lengths at most $n+2$ that are disjoint except for $\boldsymbol{s}$.
(Proof) From Lemmas 5 and 7. □

Because an n-dimensional locally twisted cube consists of two $(n-1)$-dimensional hypercubes with bijective edges between them, it is trivial that we can apply B-N2N and B-N2S in the locally twisted cube with the same performance.

Definition 4: An n-dimensional locally twisted cube, LT_n $(n \ge 3)$, is an undirected graph whose node set is $\{0,1\}^n$. Given a node $\boldsymbol{u} = (u_n, u_{n-1}, \ldots, u_1)$ in LT_n, it has n neighboring nodes $\boldsymbol{u}^{(i)}$ $(1 \le i \le n)$ where $\boldsymbol{u}^{(i)} = (u_n, u_{n-1}, \ldots, (u_{i+1} \oplus u_n), \overline{u_i}, u_{i-1}, \ldots, u_1)$ $(1 \le i \le n-2)$, $\boldsymbol{u}^{(n-1)} = (u_n, \overline{u_{n-1}}, u_{n-2}, \ldots, u_1)$, and $\boldsymbol{u}^{(n)} = (\overline{u_n}, u_{n-1}, \ldots, u_1)$. □

Theorem 3: For a source node $\boldsymbol{s}$ and a destination node $\boldsymbol{d}$ in an n-dimensional locally twisted cube with $n \ge 5$, the B-N2N algorithm takes $O(n^2)$ time and it generates n paths from $\boldsymbol{s}$ to $\boldsymbol{d}$ of lengths at most $n+2$ that are disjoint except for $\boldsymbol{s}$ and $\boldsymbol{d}$.
(Proof) From Theorem 1. □

Theorem 4: For a source node $\boldsymbol{s}$ and a set of n destination nodes $\{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_n\}$ in an n-dimensional locally twisted cube with $n \ge 5$, the B-N2S algorithm takes $O(n^2 \log n)$ time and it generates n paths from $\boldsymbol{s}$ to $\boldsymbol{d}_i$ $(1 \le i \le n)$ of lengths at most $n+2$ that are disjoint except for $\boldsymbol{s}$.
(Proof) From Theorem 2. □

6. Conclusion

In this paper, we have proposed two algorithms, B-N2N and B-N2S, that solve the node-to-node and node-to-set disjoint paths problems in the bicube, respectively. We have proved the correctness of the algorithms. We have also proved that the time complexities of the B-N2N and B-N2S algorithms are $O(n^2)$ and $O(n^2 \log n)$, respectively, and the maximum path lengths are both $n + 2$ if they are applied in an n-dimensional bicube with $n \geq 5$. In addition, we have shown that the algorithms can be applied in the locally twisted cube with the same performance.

One of our future works is to check whether the bound of the path lengths $n + 2$ is tight or not. Also, our future works include inventing an algorithm to solve the set-to-set disjoint paths problem in the bicube.

Acknowledgment

We would like to express our sincere gratitude to reviewer B for pointing out insufficient descriptions and suggesting improvements. This study was partly supported by JSPS KAKENHI Grant Number JP23K11029.

References

[1] C. L. Seitz, "The cosmic cube," *Communications of the ACM*, vol. 28, pp. 22–33, Jan. 1985.

[2] H.-S. Lim, J.-H. Park, and H.-C. Kim, "The bicube: An interconnection of two hypercubes," *International Journal of Computer Mathematics*, vol. 92, pp. 29–40, Jan. 2015.

[3] M. Okada and K. Kaneko, "Minimal paths in a bicube," *IEICE Transactions on Information and Systems*, vol. E105-D, pp. 1383–1392, Aug. 2022.

[4] H. Zhuang, W. Guo, X.-Y. Li, X. Liu, and C.-K. Lin, "The component connectivity, component diagnosability, and t/k-diagnosability of bicube networks," *Theoretical Computer Science*, vol. 896, pp. 145–157, 2021.

[5] Y.-H. Chen, S.-M. Tang, K.-J. Pai, and J.-M. Chang, "Constructing dual-cists with short diameters using a generic adjustment scheme on bicubes," *Theoretical Computer Science*, vol. 878-879, pp. 102–112, 2021.

[6] J. Liu, S. Zhou, E. Cheng, G. Chen, and M. Li, "Reliability evaluation of bicube-based multiprocessor system under the g-good-neighbor restriction," *Parallel Processing Letters*, vol. 31, no. 04, p. 2150018, 2021.

[7] J. Liu, S. Zhou, Z. Gu, Q. Zhou, and D. Wang, "Fault diagnosability of bicube networks under the PMC diagnostic model," *Theoretical Computer Science*, vol. 851, pp. 14–23, 2021.

[8] K. Kaneko, "Node-disjoint paths problems in directed bijective connection graphs," *IEICE Transactions on Information and Systems*, vol. E103-D, pp. 93–100, Jan. 2020.

[9] A. Bossard and K. Kaneko, "Time optimal node-to-set disjoint paths routing in hypercubes," *Journal of Information Science and Engineering*, vol. 30, pp. 1087–1093, July 2014.

[10] X. Yang, D. J. Evans, and G. M. Megson, "The locally twisted cubes," *International Journal of Computer Mathematics*, vol. 82, pp. 401–413, Apr. 2005.

[11] Y. Saad and M. H. Schultz, "Topological properties of hypercubes," *IEEE Transactions on Computers*, vol. 37, pp. 867–872, July 1988.

[12] P. D. Kulasinghe, "Connectivity of the crossed cube," *Information Processing Letters*, vol. 61, pp. 221–226, Feb. 1997.

[13] C.-P. Chang, J.-N. Wang, and L.-H. Hsu, "Topological properties of twisted cube," *Information Sciences*, vol. 113, pp. 147–167, Jan. 1999.

[14] Y.-K. Shih and S.-S. Kao, "One-to-one disjoint path covers on k-ary n-cubes," *Theoretical Computer Science*, vol. 412, pp. 4513–4530, Aug. 2011.

[15] D. Kocík and K. Kaneko, "Node-to-node disjoint paths problem in a Möbius cube," *IEICE Transactions on Information and Systems*, vol. E100-D, pp. 1837–1843, Aug. 2017.

[16] H. Nagashima, K. Mouri, and K. Kaneko, "Node-to-node disjoint paths in twisted crossed cubes," in *Proceedings of the 10th International Conference on Advances in Information Technology*, pp. 5:1–5:8, Dec. 2018.

[17] Q.-P. Gu and S. Peng, "Node-to-set disjoint paths problem in star graphs," *Information Processing Letters*, vol. 62, pp. 201–207, Apr. 1997.

[18] Q.-P. Gu, S. Okawa, and S. Peng, "Set-to-set fault tolerant routing in hypercudes," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E79-A, pp. 483–488, Apr. 1996.

[19] C.-N. Lai, G.-H. Chen, and D.-R. Duh, "Constructing one-to-many disjoint paths in folded hypercubes," *IEEE Transactions on Computers*, vol. 51, pp. 33–45, Jan. 2002.

[20] K. Kaneko, "An algorithm for node-to-set disjoint paths problem in bi-rotator graphs," *IEICE Transactions on Information and Systems*, vol. E89-D, pp. 647–653, Feb. 2006.

[21] Y. Suzuki, K. Kaneko, and M. Nakamori, "Node-disjoint paths in a transposition graph," *IEICE Transactions on Information and Systems*, vol. E89-D, pp. 2600–2605, Oct. 2006.

[22] D. Kocík, Y. Hirai, and K. Kaneko, "Node-to-set disjoint paths problem in a Möbius cube," *IEICE Transactions on Information and Systems*, vol. E99-D, pp. 708–713, Mar. 2016.

[23] D. Choi and I. Chung, "A nearly optimal one-to-many routing algorithm on k-ary n-cube networks," *Smart Media Journal*, vol. 7, pp. 9–14, May 2018.

[24] X.-B. Chen, "Many-to-many disjoint paths in faulty hypercubes," *Information Sciences*, vol. 179, pp. 3110–3115, Aug. 2009.

[25] X.-J. Li, B. Liu, M. Ma, and J.-M. Xu, "Many-to-many disjoint paths in hypercubes with faulty vertices," *Discrete Applied Mathematics*, vol. 217, pp. 229–242, Jan. 2017.

[26] J. Li, C. Melekian, S. Zuo, and E. Cheng, "Unpaired many-to-many disjoint path covers on bipartite k-ary n-cube networks with faulty elements," *International Journal of Foundations of Computer Science*, vol. 31, no. 03, pp. 371–383, 2020.

[27] H. Ichida and K. Kaneko, "Set-to-set disjoint paths problem in Möbius cubes," *IEEE Access*, vol. 10, pp. 83075–83084, 2022.

[28] K. Kaneko, A. Bossard, and F. C. Harris, Jr., "Set-to-set disjoint path routing in bijective connection graphs," *IEEE Access*, vol. 10, pp. 72731–72742, 2022.

Arata Kaneko is a master program student at Tokyo University of Agriculture and Technology in Japan. His main research areas are interconnection networks and fault-tolerant systems based on graph theory and network theory. He received the B.E. degree from Tokyo University of Agriculture and Technology in 2022.

Htoo Htoo Sandi Kyaw is an Assistant Professor at Tokyo University of Agriculture and Technology in Japan. Her main research areas are educational technology, web application systems, and graph theory. She received the B.E. and M.E. degrees from University of Technology (Yatanarpon Cyber City) in Myanmar in 2015 and 2018, respectively, and the Ph.D. degree from Okayama University in Japan in 2021.

Kunihiro Fujiyoshi is an Associate Professor at Tokyo University of Agriculture and Technology in Japan. His main research interests are in combinatorial algorithms and VLSI layout design. He received the B.E., M.E., and D.E. degrees from Tokyo Institute of Technology in 1987, 1989, and 1994, respectively. He is a member of IEEE and IPSJ.

Keiichi Kaneko is a Professor at Tokyo University of Agriculture and Technology in Japan. His main research areas are functional programming, parallel and distributed computation, partial evaluation and fault-tolerant systems. He received the B.E., M.E., and Ph.D. degrees from the University of Tokyo in 1985, 1987, and 1994, respectively. He is a member of ACM, IEEE CS, IPSJ and JSSST.