



# Node-to-set disjoint paths problem in cross-cubes

Xi Wang<sup>1,2</sup> · Jianxi Fan<sup>2</sup> · Shukui Zhang<sup>2</sup> · Jia Yu<sup>3</sup>

Accepted: 6 May 2021 / Published online: 9 June 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

## Abstract

Hypercubes are popular topologies of massive multiprocessor systems due to their super properties. Cross-cubes are significant variations of hypercubes and they have smaller diameters and higher fault-tolerant capability than hypercubes at the same dimensions. In this paper, we construct node-to-set disjoint paths of an  $n$ -dimensional cross-cube,  $C_n$ , whose maximum length is limited by  $2n - 3$ . Furthermore, we propose an  $O(M \log^2 N)$  algorithm with a view to finding node-to-set disjoint paths of  $C_n$ , where  $N$  is the node number of  $C_n$ . And we also present the simulation results for the maximal length of disjoint paths obtained by our algorithm.

**Keywords** Interconnection networks · Cross-cube · Node-to-set disjoint paths · Parallel processing

## 1 Introduction

Huge multiprocessor systems are solutions to large-scale computing problems such as weather forecasting, scientific research, intelligence agencies, and data mining [1]. Hypercubes are widely used in interconnection networks of massively multiprocessor systems due to their excellent properties, including regularity, symmetry, and small link complexity [2]. Based on the constructive method of hypercubes, many variants of hypercubes were proposed [3–6]. The  $n$ -dimensional cross-cube,  $C_n$ , has smaller diameter and higher fault-tolerant capability than the  $n$ -dimensional hypercube,  $Q_n$ . Moreover, a cycle of odd length and a complete binary tree of height  $n$  can be embedded into  $C_n$ , while  $Q_n$  does not have these properties [3].

The disjoint path problems are substantial in the design and implementation of interconnection networks [7–9]. The node-to-set disjoint paths problem is one of the

---

✉ Shukui Zhang  
zhangsk@suda.edu.cn

<sup>1</sup> Suzhou Institute of Industrial Technology, Suzhou 215004, China

<sup>2</sup> School of Computer Science and Technology, Soochow University, Suzhou 215006, China

<sup>3</sup> College of Computer Science and Technology, Qingdao University, Qingdao 266071, China

disjoint path problems: Given a single source node  $s$  and a set  $D = \{d_1, d_2, \dots, d_k\}$  of  $k$  destination nodes with  $s \notin D$  in a  $k$ -connected graph  $G = (V(G), E(G))$ , construct  $k$  paths from  $s$  to  $d_i$  which are node-disjoint except for  $s$  for all  $1 \leq i \leq k$ . As the number of nodes in interconnection networks increases, node failure seems to become the norm rather than exception. Therefore, constructing disjoint paths increases the probability of finding fault-free communication route in interconnection networks. So far, problems of finding node-to-set disjoint paths on some classes of graphs have been studied, such as hypercubes [10], Möbius cubes [11], star graphs [12], pancake graphs [13], burnt pancake graphs [14], hierarchical cubic networks [15], perfect hierarchical hypercubes [16, 17], and biswapped networks [18].

In general, the node-to-set disjoint path problems can be obtained from the polynomial-order of  $|V(G)|$  by applying the maximum flow algorithm. However, the complexity of the algorithm is not eligible for an  $n$ -dimensional cross-cube  $C_n$  because the number of nodes in  $C_n$  is  $2^n$ . Bossard and Kaneko proposed an  $O(\log^2 N)$  algorithm to find node-to-set disjoint paths in an  $n$ -dimensional hypercube  $Q_n$ , where their longest length is no more than  $n + 1$  and  $N$  is the node number of  $Q_n$  [10]. Unfortunately, this algorithm is not appropriate for cross-cubes directly because they do not exist some properties that hold in hypercubes. Therefore, it is necessary to design an applicable algorithm by studying properties of a  $C_n$ . In this paper, we propose an  $O(N \log^2 N)$  algorithm for constructing  $n + 1$  node-to-set disjoint paths in  $C_n$ , where  $n + 1$  is equivalent to the connectivity of  $C_n$ 's and  $N$  is the node number of  $C_n$ . Moreover, we prove that the length of the longest path of the  $n + 1$  disjoint paths constructed by our algorithm is no more than  $2n - 3$ . We also show the experiments of  $C_n$  to evaluate the performance of the algorithm we have proposed.

The remainder of this paper is organized as follows. Section 2 provides relevant preliminary knowledge. In Sect. 3, the construction of node-to-set disjoint paths in cross-cubes is studied. Conclusions of this paper and suggestions on future work are given in Sect. 4.

## 2 Preliminaries

For terminology and notation not defined here, we follow [19]. Given a simple graph  $G = G(V(G), E(G))$  to represent an interconnection network, where  $V(G)$  and  $E(G)$  denote the node set and edge set, respectively. A path in a graph is a sequence of nodes,  $P = (u_1, u_2, \dots, u_j, \dots, u_{n-1}, u_n)$ , in which no node is repeated and  $u_j, u_{j+1}$  are adjacent for any integer  $1 \leq j < n$ . The length of a path  $P$ ,  $l(P)$ , is the number of edges in  $P$ , and we use  $V(P)$  to represent the node set in  $P$ . We also write the path  $(u_1, u_2, \dots, u_k)$  as  $(u_1, Q, u_i, u_{i+1}, \dots, u_k)$ , where  $Q$  is the path  $(u_2, u_3, \dots, u_{i-1})$ . If  $u$  and  $v$  are nodes on a path  $P$ , we write  $\text{Path}(P, u, v)$  to denote the sub-path of  $P$  from  $u$  to  $v$ . Furthermore, we use  $P[i]$  to denote the node  $u_i$  for  $P = (u_1, u_2, \dots, u_n)$  for any integer  $1 \leq i \leq n$  and use  $P[-1]$  to denote the last node in  $P$ .

For  $U \subseteq V(G)$ , we use  $G[U] = (U, E')$  to denote the subgraph induced by  $U$  in  $G$  where  $E' = \{(u, v) \in E(G) | u, v \in U\}$ . Given two distinct nodes  $u$  and  $v$  of  $G$ , the distance between  $u$  and  $v$  is defined as the length of the shortest path between  $u$  and  $v$  in  $G$ , denoted by  $\text{dist}(G, u, v)$ . The diameter of  $G$  is defined as  $d(G) =$

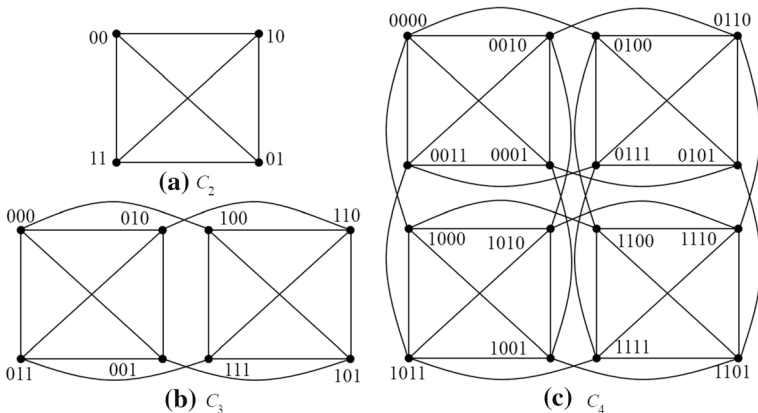
$\max(\text{dist}(G, u, v) | u, v \in V(G), u \neq v)$ . A graph is connected when there is a path between every pair of nodes. In a connected graph, there are no unreachable nodes. A graph that is not connected is called to be disconnected. The connectivity  $\kappa(G)$  of a connected graph  $G$  (other than a complete graph) is the minimum number of nodes whose removal disconnects  $G$ . We say  $u$  is a neighbor of  $v$  or  $u$  is adjacent to  $v$  if  $(u, v) \in E(G)$ . Then, we use  $N_G(u)$  to denote neighbors of  $u$  in graph  $G$ .

A binary string  $u$  with length  $n$  can be written as  $u_{n-1}u_{n-2} \dots u_0$  and  $u[i] = u_i \in \{0, 1\}$  is denoted by the  $i$ -th bit of  $u$  for any integer  $i$  with  $0 \leq i \leq n - 1$ . The complement of  $u_i$  is denoted by  $\bar{u}_i$  or  $1 - u_i$ .

The  $n$ -dimensional cross-cube,  $C_n$ , has  $2^n$  nodes and  $(n + 1)2^{n-1}$  edges. Each node of  $C_n$  is represented by a unique binary string from 0 to  $2^n - 1$  of length  $n$ , called the address of the node. For  $i \in \{0, 1\}$ , let  $C_{n-1}^i$  denote the graph obtained by prefixing the address of each node of  $C_{n-1}$  with  $i$ . Obviously,  $C_{n-1}^i$  is isomorphic to  $C_{n-1}$  with  $i \in \{0, 1\}$ . In this paper, we would not make a distinction between nodes of  $C_n$  and their addresses. We adopt the recursively definitions of  $C_n$  from [3] as follows.

**Definition 1**  $C_2$  is a complete graph on four nodes whose addresses are 00, 01, 10, and 11. For  $n \geq 3$ ,  $C_n$  consists of  $C_{n-1}^0$  and  $C_{n-1}^1$ . The nodes  $u = u_{n-1}u_{n-2} \dots u_0 \in V(C_{n-1}^0)$  and  $v = v_{n-1}v_{n-2} \dots v_0 \in V(C_{n-1}^1)$  are joined by an edge in  $C_n$  if and only if  $u_{n-2}u_{n-3} \dots u_0 = v_{n-2}v_{n-3} \dots v_0$ .

Figure 1 depicts several cross-cubes such as  $C_2, C_3$ , and  $C_4$ . For  $n \geq 3$ , let  $u = u_{n-1}u_{n-2} \dots u_1u_0 \in V(C_n)$ . Then, let  $(u)^2 = \{u_{n-1}u_{n-2} \dots u_2\alpha | \alpha \in \{00, 01, 10, 11\} \text{ and } \alpha \neq u_1u_0\}$  be 2-neighbors of  $u$  and  $(u)^i = u_{n-1}u_{n-2} \dots (1 - u_{i-1}) \dots u_1u_0$  be  $i$ -neighbor of  $u$  for  $3 \leq i \leq n$ .



**Fig. 1** The structure of **a** two-dimensional cross-cube  $C_2$ , **b** three-dimensional cross-cube  $C_3$ , and **c** four-dimensional cross-cube  $C_4$ , respectively

**Lemma 2** *An  $n$ -dimensional cross-cube  $C_n$  has the following combinatorial properties [3].*

1.  $C_n$  is  $(n + 1)$ -regular;
2. The diameter of  $C_n$  is  $d(C_n) = n - 1$ ;
3. The connectivity of  $C_n$  is  $\kappa(C_n) = n + 1$ .

**Definition 3** The node-to-set disjoint paths problem in an  $n$ -dimensional cross-cube is to construct  $n + 1$  paths from a single source node  $s$  to each node in a destination nodes set  $D = \{d_1, d_2, \dots, d_{n+1}\}$  which are node-disjoint except for  $s$ .

### 3 Main results

In this section, we construct  $n + 1$  node-to-set disjoint paths in  $C_n$  in Theorem 10. Then, we propose an  $O(N \log^2 N)$  algorithm for finding  $n + 1$  node-to-set disjoint paths in  $C_n$ , where  $N$  is the node number of  $C_n$ . Furthermore, we show simulations and experiments of  $C_n$  to evaluate the performance of our algorithm proposed in cross-cubes.

#### 3.1 Node-to-set disjoint paths in cross-cubes

In this subsection, one of main results is Theorem 10, in which the one-to-set disjoint paths of  $C_n$  are constructed for any integer  $n \geq 2$ , using mathematical induction on  $n$ . For this purpose, Lemmas 4–9 are presented for the following construction schemes.

In [20], Wang, He, and Zhang proposed an  $O(n)$  algorithm to find a shortest path between any two distinct nodes  $u$  and  $v$  in  $C_n$ , which is called CRouting( $C_n, u, v$ ) algorithm. By Definition 1 and Lemma 2, the following lemma holds.

**Lemma 4**  $l(\text{CRouting}(C_n, u, v)) \leq d(C_n) = n - 1$ .

**Lemma 5** *For any integer  $n \in \{2, 3\}$ , choose a source node  $s$  and an arbitrary node set  $D = \{d_1, d_2, \dots, d_{n+1}\}$  in a  $C_n$  with  $s \notin D$ , there exist  $n + 1$  node-disjoint paths (except for  $s$ ) from  $s$  to nodes of  $D$ , whose maximum length is limited by  $2n - 3$ .*

**Proof** First of all, we show that the lemma holds for  $n = 2$  since  $C_2$  is a complete graph with 4 nodes [19]. Secondly, the lemma is verified by a computer program when  $n = 3$ . For example, the four paths  $P_1^1, P_1^2, P_1^3, P_1^4$  (resp.  $P_2^1, P_2^2, P_2^3, P_2^4; P_3^1, P_3^2, P_3^3, P_3^4; \dots; P_{35}^1, P_{35}^2, P_{35}^3, P_{35}^4$ ) from  $s = 000$  to nodes of  $D = \{010, 011, 001, 111\}$  (resp.  $\{010, 011, 001, 110\}, \{010, 011, 001, 100\}, \{010, 011, 001, 101\}, \{010, 011, 111, 110\}, \{010, 011, 111, 100\}, \{010, 011, 111, 101\}, \{010, 011, 110, 100\}, \{010, 011, 110, 101\}, \{010, 011, 100, 101\}, \{010, 001, 111, 110\}, \{010, 001, 111, 100\}, \{010, 001, 111, 101\}, \{010, 001, 110, 100\}, \{010, 001, 110, 101\}, \{010, 001, 100, 101\}, \{010, 111, 110, 100\}, \{010, 111, 110, 101\}, \{010, 111, 100, 101\}, \{011, 001, 111, 110\}$ ),

$\{011, 001, 111, 100\}, \{011, 001, 111, 101\}, \{011, 001, 110, 100\}, \{011, 001, 110, 101\},$   
 $\{011, 001, 100, 101\}, \{011, 111, 110, 100\}, \{011, 111, 110, 101\}, \{011, 111, 100, 101\},$   
 $\{011, 110, 100, 101\}, \{001, 111, 110, 100\}, \{001, 111, 110, 101\}, \{001, 111, 100, 101\},$   
 $\{001, 110, 100, 101\}, \{111, 110, 100, 101\}$ ) are node-disjoint (except for  $s$ ), whose maximum length is limited by 3, are listed in Table 1.  $\square$

**Lemma 6** For any integer  $n \geq 3, i \in \{0, 1\}$ , and any node  $u \in V(C_{n-1}^i)$ , let  $u' = (u)^n$  and  $x_1, x_2, \dots, x_n$  be the  $n$  neighbors of  $u$  in  $C_{n-1}^i$ . Furthermore, let  $x'_j = (x_j)^n$  for all  $j \in \{1, 2, \dots, n\}$ . Then,  $P_1 = (u, x_1, x'_1), P_2 = (u, x_2, x'_2), \dots, P_n = (u, x_n, x'_n), P_{n+1} = (u, u')$  are  $n + 1$  node-disjoint paths (except for  $u$ ) from  $u$  into  $C_{n-1}^{1-i}$ .

**Proof** By Definition 1, for any integers  $j$  and  $k$  with  $1 \leq j, k \leq n$  and  $j \neq k$ , we have  $x_j \neq x_k$  and hence  $x'_j \neq x'_k$ . Then, since  $u \neq x_j$  for any integer  $j$  with  $1 \leq j \leq n$ , so  $u' \neq x'_j$ . Accordingly,  $P_1 = (u, x_1, x'_1), P_2 = (u, x_2, x'_2), \dots, P_n = (u, x_n, x'_n), P_{n+1} = (u, u')$  are  $n + 1$  node-disjoint paths (except for  $u$ ) from  $u$  into  $C_{n-1}^{1-i}$  (refer to Fig. 2).  $\square$

**Lemma 7** For any integer  $n \geq 3, i \in \{0, 1\}$ , and any node  $u \in V(C_{n-1}^i)$ , let  $u' = (u)^n$ . For any integer  $k$  with  $1 \leq k \leq n - 1$ , let  $T = \{u', y_1, y_2, \dots, y_k, y'_1, y'_2, \dots, y'_k\}$  with  $u \notin T, y_1, y_2, \dots, y_k \in V(C_{n-1}^i)$ , and  $y'_1, y'_2, \dots, y'_k \in V(C_{n-1}^{1-i})$  such that  $y'_j = (y_j)^n$  for all  $j \in \{1, 2, \dots, k\}$ . Then, there exists a path  $P = (u, x, x')$  such that  $(u, x) \in E(C_{n-1}^i), x' = (x)^n$ , and  $V(P) \cap T = \emptyset$ .

**Proof** Let  $x_1, x_2, \dots, x_n$  be the  $n$  neighbors of  $u$  in  $C_{n-1}^i$ . Furthermore, let  $x'_j = (x_j)^n$  for all  $j \in \{1, 2, \dots, n\}$ . By Lemma 6, these exist  $n + 1$  node-disjoint paths (except for  $u$ )  $P_1 = (u, x_1, x'_1), P_2 = (u, x_2, x'_2), \dots, P_n = (u, x_n, x'_n), P_{n+1} = (u, u')$  from  $u$  into  $C_{n-1}^{1-i}$ . Accordingly, we have  $V(P_{n+1}) \cap T = \{u'\}$ . Since  $k \leq n - 1$ , there exists a node  $x \in \{x_1, x_2, \dots, x_n\}$  such that  $x \notin \{y_1, y_2, \dots, y_k\}$  and  $(u, x) \in E(C_{n-1}^i)$ . Let  $x' = (x)^n$  and  $P = (u, x, x')$  (refer to Fig. 3). By Definition 1, we can verify that  $x \notin T$  and  $x' \notin \{y'_1, y'_2, \dots, y'_k, u'\}$ . Hence, we have  $V(P) \cap T = \emptyset$ .  $\square$

**Lemma 8** For any integer  $n \geq 3, i \in \{0, 1\}$ , and  $u_1, u_2, \dots, u_r \in V(C_{n-1}^i)$  with  $1 \leq r \leq \lceil \frac{n-1}{2} \rceil$ , let  $u'_j = (u_j)^n$  for  $j \in \{1, 2, \dots, r\}$ . For any integer  $k$  with  $1 \leq k \leq n - 2r + 1$ , let  $T = \{u'_1, u'_2, \dots, u'_r, y_1, y_2, \dots, y_k, y'_1, y'_2, \dots, y'_k\}$  with  $\{u_1, u_2, \dots, u_r\} \cap T = \emptyset$  such that  $y_j \in V(C_{n-1}^i), y'_j \in V(C_{n-1}^{1-i})$ , and  $y'_j = (y_j)^n$  for  $j \in \{1, 2, \dots, k\}$ . Then,

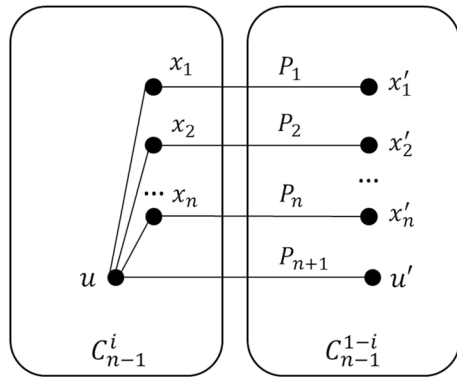
- (1) if  $r = 1$ , there exists a path  $P = (u_1, x_1, z_1)$  such that  $(u_1, x_1) \in E(C_{n-1}^i), z_1 = (x_1)^n$ , and  $V(P) \cap T = \emptyset$ .

**Table 1** Four node-disjoint paths from 000 to nodes of  $D$  in  $C_3$ 

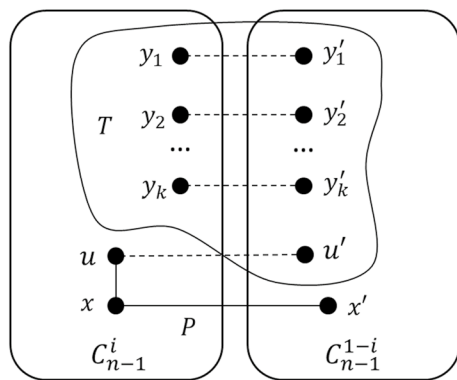
$D = \{010, 011, 001, 111\}$	$P_1^1 = (000, 010), P_1^2 = (000, 011), P_1^3 = (000, 001), P_1^4 = (000, 100, 111)$
$D = \{010, 011, 001, 110\}$	$P_2^1 = (000, 010), P_2^2 = (000, 011), P_2^3 = (000, 001), P_2^4 = (000, 100, 110)$
$D = \{010, 011, 001, 100\}$	$P_3^1 = (000, 010), P_3^2 = (000, 011), P_3^3 = (000, 001), P_3^4 = (000, 100)$
$D = \{010, 011, 001, 101\}$	$P_4^1 = (000, 010), P_4^2 = (000, 011), P_4^3 = (000, 001), P_4^4 = (000, 100, 101)$
$D = \{010, 011, 111, 110\}$	$P_5^1 = (000, 010), P_5^2 = (000, 011), P_5^3 = (000, 100, 111), P_5^4 = (000, 001, 101, 110)$
$D = \{010, 011, 111, 100\}$	$P_6^1 = (000, 010), P_6^2 = (000, 011), P_6^3 = (000, 001, 101, 111), P_6^4 = (000, 100)$
$D = \{010, 011, 111, 101\}$	$P_7^1 = (000, 010), P_7^2 = (000, 011), P_7^3 = (000, 100, 111), P_7^4 = (000, 001, 101)$
$D = \{010, 011, 110, 100\}$	$P_8^1 = (000, 010), P_8^2 = (000, 011), P_8^3 = (000, 001, 101, 110), P_8^4 = (000, 100)$
$D = \{010, 011, 110, 101\}$	$P_9^1 = (000, 010), P_9^2 = (000, 011), P_9^3 = (000, 100, 110), P_9^4 = (000, 001, 101)$
$D = \{010, 011, 100, 101\}$	$P_{10}^1 = (000, 010), P_{10}^2 = (000, 011), P_{10}^3 = (000, 100), P_{10}^4 = (000, 001, 101)$
$D = \{010, 001, 111, 110\}$	$P_{11}^1 = (000, 010), P_{11}^2 = (000, 001), P_{11}^3 = (000, 011, 111), P_{11}^4 = (000, 100, 110)$
$D = \{010, 001, 111, 100\}$	$P_{12}^1 = (000, 010), P_{12}^2 = (000, 001), P_{12}^3 = (000, 011, 111), P_{12}^4 = (000, 100)$
$D = \{010, 001, 111, 101\}$	$P_{13}^1 = (000, 010), P_{13}^2 = (000, 001), P_{13}^3 = (000, 011, 111), P_{13}^4 = (000, 100, 101)$
$D = \{010, 001, 110, 100\}$	$P_{14}^1 = (000, 010), P_{14}^2 = (000, 001), P_{14}^3 = (000, 011, 111, 110), P_{14}^4 = (000, 100)$
$D = \{010, 001, 110, 101\}$	$P_{15}^1 = (000, 010), P_{15}^2 = (000, 001), P_{15}^3 = (000, 100, 110), P_{15}^4 = (000, 011, 111, 101)$
$D = \{010, 001, 100, 101\}$	$P_{16}^1 = (000, 010), P_{16}^2 = (000, 001), P_{16}^3 = (000, 100), P_{16}^4 = (000, 011, 111, 101)$
$D = \{010, 111, 110, 100\}$	$P_{17}^1 = (000, 010), P_{17}^2 = (000, 011, 111), P_{17}^3 = (000, 001, 101, 110), P_{17}^4 = (000, 100)$
$D = \{010, 111, 110, 101\}$	$P_{18}^1 = (000, 010), P_{18}^2 = (000, 011, 111), P_{18}^3 = (000, 100, 110), P_{18}^4 = (000, 001, 101)$
$D = \{010, 111, 100, 101\}$	$P_{19}^1 = (000, 010), P_{19}^2 = (000, 011, 111), P_{19}^3 = (000, 100), P_{19}^4 = (000, 001, 101)$
$D = \{010, 110, 100, 101\}$	$P_{20}^1 = (000, 010), P_{20}^2 = (000, 011, 111, 110), P_{20}^3 = (000, 100), P_{20}^4 = (000, 001, 101)$
$D = \{011, 001, 111, 110\}$	$P_{21}^1 = (000, 011), P_{21}^2 = (000, 001), P_{21}^3 = (000, 100, 111), P_{21}^4 = (000, 010, 110)$
$D = \{011, 001, 111, 100\}$	$P_{22}^1 = (000, 011), P_{22}^2 = (000, 001), P_{22}^3 = (000, 010, 110, 111), P_{22}^4 = (000, 100)$
$D = \{011, 001, 111, 101\}$	$P_{23}^1 = (000, 011), P_{23}^2 = (000, 001), P_{23}^3 = (000, 100, 111), P_{23}^4 = (000, 010, 110, 101)$
$D = \{011, 001, 110, 100\}$	$P_{24}^1 = (000, 011), P_{24}^2 = (000, 001), P_{24}^3 = (000, 010, 110), P_{24}^4 = (000, 100)$
$D = \{011, 001, 110, 101\}$	$P_{25}^1 = (000, 011), P_{25}^2 = (000, 001), P_{25}^3 = (000, 010, 110), P_{25}^4 = (000, 100, 101)$
$D = \{011, 001, 100, 101\}$	$P_{26}^1 = (000, 011), P_{26}^2 = (000, 001), P_{26}^3 = (000, 100), P_{26}^4 = (000, 010, 110, 101)$
$D = \{011, 111, 110, 100\}$	$P_{27}^1 = (000, 011), P_{27}^2 = (000, 001, 101, 111), P_{27}^3 = (000, 010, 110), P_{27}^4 = (000, 100)$
$D = \{011, 111, 110, 101\}$	$P_{28}^1 = (000, 011), P_{28}^2 = (000, 100, 111), P_{28}^3 = (000, 010, 110), P_{28}^4 = (000, 001, 101)$
$D = \{011, 111, 100, 101\}$	$P_{29}^1 = (000, 011), P_{29}^2 = (000, 010, 110, 111), P_{29}^3 = (000, 100), P_{29}^4 = (000, 001, 101)$
$D = \{011, 110, 100, 101\}$	$P_{30}^1 = (000, 011), P_{30}^2 = (000, 010, 110), P_{30}^3 = (000, 100), P_{30}^4 = (000, 001, 101)$
$D = \{001, 111, 110, 100\}$	$P_{31}^1 = (000, 001), P_{31}^2 = (000, 011, 111), P_{31}^3 = (000, 010, 110), P_{31}^4 = (000, 100)$
$D = \{001, 111, 110, 101\}$	$P_{32}^1 = (000, 001), P_{32}^2 = (000, 011, 111), P_{32}^3 = (000, 010, 110), P_{32}^4 = (000, 100, 101)$
$D = \{001, 111, 100, 101\}$	$P_{33}^1 = (000, 001), P_{33}^2 = (000, 011, 111), P_{33}^3 = (000, 100), P_{33}^4 = (000, 010, 110, 101)$
$D = \{001, 110, 100, 101\}$	$P_{34}^1 = (000, 001), P_{34}^2 = (000, 010, 110), P_{34}^3 = (000, 100), P_{34}^4 = (000, 011, 111, 101)$
$D = \{111, 110, 100, 101\}$	$P_{35}^1 = (000, 011, 111), P_{35}^2 = (000, 010, 110), P_{35}^3 = (000, 100), P_{35}^4 = (000, 001, 101)$

- (2) if  $2 \leq r \leq \lceil \frac{n-1}{2} \rceil$ , there exist  $r$  node-disjoint paths  $P_1 = (u_1, x_1, z_1)$ ,  $P_2 = (u_2, x_2, z_2), \dots, P_r = (u_r, x_r, z_r)$  such that  $(u_j, x_j) \in E(C_{n-1}^i)$ ,  $z_j = (x_j)^n$ , and  $V(P_j) \cap T = \emptyset$  for  $j \in \{1, 2, \dots, r\}$ .

**Fig. 2**  $n + 1$  node-disjoint paths (except for  $u$ ) from  $u$  into  $C_{n-1}^{1-i}$



**Fig. 3** Example of  $P = (u, x, x')$  with  $V(P) \cap T = \emptyset$



**Proof** If  $n = 3$ , we have  $r = 1$  and thus the lemma holds based on Lemma 7. In what follows, we will prove that the lemma holds for  $n \geq 4$  by induction on  $r$ . By Lemma 7, the lemma holds for  $r = 1$ . We assume that the lemma holds for  $r = \tau$  when  $1 \leq \tau \leq \lceil \frac{n-1}{2} \rceil - 1$ .

Then, we consider  $r = \tau + 1$ . Note that, we have  $2 \leq r = \tau + 1 \leq \lceil \frac{n-1}{2} \rceil$ . For any integer  $m$  with  $1 \leq m \leq n - 2(\tau + 1) + 1 < n - 2\tau + 1$ , without loss of generality, denote  $T' = \{u'_1, u'_2, \dots, u'_\tau, y_1, y_2, \dots, y_m, y'_1, y'_2, \dots, y'_m\}$ . Clearly, we have  $T' \subseteq T$  and  $\{u_1, u_2, \dots, u_\tau\} \cap T' = \emptyset$ . Accordingly, we have  $V(P_j) \cap T' = \emptyset$  for  $j \in \{1, 2, \dots, \tau\}$ .

Choose a node  $u_{\tau+1} \in V(C_{n-1}^i)$  such that  $u_{\tau+1} \notin \{u_1, u_2, \dots, u_\tau\}$  and  $u_{\tau+1} \notin T'$ . Let  $u'_{\tau+1} = (u_{\tau+1})^n$  and  $T'' = T' \cup \{u'_{\tau+1}\} = \{u'_1, u'_2, \dots, u'_\tau, u'_{\tau+1}, y_1, y_2, \dots, y_m, y'_1, y'_2, \dots, y'_m\}$ . By Definition 1, it is easy to verify that  $\{u_1, u_2, \dots, u_\tau, u_{\tau+1}\} \cap T'' = \emptyset$  and  $u'_{\tau+1} \notin \{z_1, z_2, \dots, z_\tau\}$ . Thus, we have  $V(P_j) \cap T'' = \emptyset$  for  $j \in \{1, 2, \dots, \tau\}$ .

Denote  $D = \{u_1, u_2, \dots, u_\tau\} \cup \{x_1, x_2, \dots, x_\tau\} \cup \{y_1, y_2, \dots, y_m\}$ . Then, we have

$$|D| = \tau + \tau + m \leq 2\tau + n - 2(\tau + 1) + 1 = n - 1 < n.$$

Let  $x_{\tau+1}^1, x_{\tau+1}^2, \dots, x_{\tau+1}^n$  be the  $n$  neighbors of  $u_{\tau+1}$  in  $C_{n-1}^i$ . Since  $|D| < n$ , there exists a node  $x_{\tau+1} \in \{x_{\tau+1}^1, x_{\tau+1}^2, \dots, x_{\tau+1}^n\}$  such that  $x_{\tau+1} \notin D$  and  $(u_{\tau+1}, x_{\tau+1}) \in E(C_{n-1}^i)$ . Let  $z_{\tau+1} = (x_{\tau+1})^n$  and  $P_{\tau+1} = (u_{\tau+1}, x_{\tau+1}, z_{\tau+1})$ . By Definition 1, we have  $x_{\tau+1}, z_{\tau+1} \notin T''$  and  $x_{\tau+1}, z_{\tau+1} \notin V(P_j)$  for  $j \in \{1, 2, \dots, \tau\}$ .

Hence,  $P_1 = (u_1, x_1, z_1), P_2 = (u_2, x_2, z_2), \dots, P_\tau = (u_\tau, x_\tau, z_\tau), P_{\tau+1} = (u_{\tau+1}, x_{\tau+1}, z_{\tau+1})$  are  $\tau + 1$  node-disjoint paths such that  $V(P_j) \cap T'' = \emptyset$  for  $j \in \{1, 2, \dots, \tau + 1\}$  (refer to Fig. 4).

From the above discussion, we claim that the lemma holds. □

**Lemma 9** For any integer  $n \geq 3$ , choose an arbitrary source node  $s$  and an arbitrary node set  $D = \{d_1, d_2, \dots, d_{n+1}\}$  in a  $C_n$  with  $s \notin D$ , there exist  $n + 1$  node-disjoint paths (except for  $s$ ) from  $s$  to nodes of  $D$ , whose maximum length is limited by  $2n - 3$ .

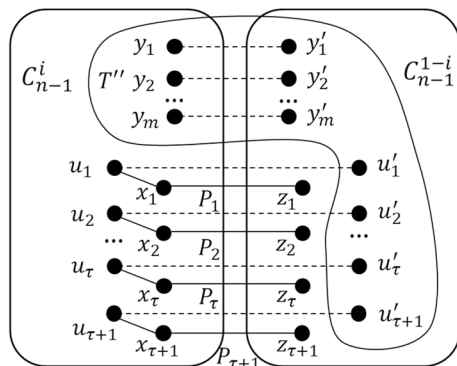
**Proof** The lemma is proved by induction on  $n$ , which is the dimension of  $C_n$ . By Lemma 5, the lemma holds for  $n = 3$ . We assume that the lemma holds for  $n = \tau - 1$  when  $\tau \geq 4$ . Then, we can obtain the following statement based on the induction hypothesis: for a node  $s$  and a node set  $D' = \{d_1, d_2, \dots, d_\tau\}$  in a  $C_{\tau-1}$  with  $s \notin D'$  and  $\tau \geq 4$ , there exist  $\tau$  node-disjoint paths (except for  $s$ ) from  $s$  to nodes of  $D'$ , whose maximum length is limited by  $2(\tau - 1) - 3 = 2\tau - 5$ . In what follows, by choosing an arbitrary source node  $s$  and an arbitrary node set  $D = \{d_1, d_2, \dots, d_{\tau+1}\}$  in a  $C_\tau$  with  $s \notin D$ , we will prove that the lemma holds for  $n = \tau$  when  $\tau \geq 4$ .

Choose an integer  $i$  with  $i \in \{0, 1\}$ . Let  $D_1 = D \cap V(C_{\tau-1}^i), D_2 = D \cap V(C_{\tau-1}^{1-i}), D_3 = \{u|u \in D_1 \text{ and } (u)^\tau \in D\}, D_4 = \{u|u \in D_2 \text{ and } (u)^\tau \in D\}, \alpha = |D_1|, \beta = |D_2|, \gamma = |D_3| = |D_4|$ , and  $s' = (s)^\tau$ . Without loss of generality, suppose that  $s \in V(C_{\tau-1}^i)$ . Then, we can claim the following three cases with respect to  $\alpha$ .

Case 1.  $\alpha = \tau + 1$ .

Let  $D'' = \{d_2, d_3, \dots, d_{\tau+1}\}$  and  $\phi = 0$ . According to induction hypothesis, there exist  $\tau$  paths node-disjoint paths  $Q_1, Q_2, \dots, Q_\tau$  (except for  $s$ ) from  $s$  to nodes of  $D'$ ,

**Fig. 4**  $\tau + 1$  node-disjoint paths  $P_1, P_2, \dots, P_\tau, P_{\tau+1}$  in  $C_n$  with  $V(P_j) \cap T'' = \emptyset$  for  $j \in \{1, 2, \dots, \tau + 1\}$





whose maximum length is limited by  $2\tau - 5$ . Choose  $\phi \in \{1, 2, \dots, \tau\}$  such that  $d_{\tau+1} \in V(Q_\phi)$ . We further obtain the following two subcases with respect to  $\phi$ .

Case 1.1  $\phi \notin \{1, 2, \dots, \tau\}$ .

Let  $d'_{\tau+1} = (d_{\tau+1})^\tau$  and  $Q' = \text{CRouting}(C_{\tau-1}^{1-i}, s', d'_{\tau+1})$ . For any integer  $1 \leq j \leq \tau + 1$ , if  $1 \leq j \leq \tau$ , let  $P_j = Q_j$ ; otherwise, let  $P_j = (s, Q', d_{\tau+1})$  (refer to Fig. 5a).

By the induction hypothesis,  $P_1, P_2, \dots, P_\tau$  are node-disjoint expect for  $s$ . In addition, we can verify that  $V(P_{\tau+1}) \cap V(P_k) = \{s\}$  for all  $1 \leq k \leq \tau$ . Therefore, we state that  $P_1, P_2, \dots, P_{\tau+1}$  are node-disjoint expect for  $s$ . By Lemma 4, we have

$$|P_j| \leq \max\{2\tau - 5, \tau - 2 + 2\} \leq 2\tau - 3$$

for  $j \in \{1, 2, \dots, \tau + 1\}$  with  $\tau \geq 4$ . From these discussions,  $\tau + 1$  paths  $P_1, P_2, \dots, P_{\tau+1}$  joining  $s$  and nodes of  $D$  in  $C_\tau$  are node-disjoint expect for  $s$ , which is consistent with our assumptions.

Subcase 1.2  $\phi \in \{1, 2, \dots, \tau\}$ .

Without loss of generality, denote  $\phi = 1$ . Let  $d'_1 = (d_1)^\tau$  and  $Q' = \text{CRouting}(C_{\tau-1}^{1-i}, s', d'_1)$ . For any integer  $1 \leq j \leq \tau + 1$ , let

$$P_j = \begin{cases} (s, Q', d_1) & \text{if } j = 1, \\ Q_j & \text{if } 2 \leq j \leq \tau, \\ \text{Path}(Q_1, s, d_{\tau+1}) & \text{if } j = \tau + 1. \end{cases}$$

Accordingly, all the  $\tau + 1$  needed paths connecting  $s$  and nodes of  $D = \{d_1, d_2, \dots, d_{\tau+1}\}$  of  $C_\tau$  are constructed above (refer to Fig. 5b). By the induction hypothesis,  $P_2, P_3, \dots, P_{\tau+1}$  are node-disjoint expect for  $s$ . In addition, we can verify that  $V(P_1) \cap V(P_k) = \{s\}$  for all  $2 \leq k \leq \tau + 1$ . Therefore, we state that  $P_1, P_2, \dots, P_{\tau+1}$  are node-disjoint expect for  $s$ . By Lemma 4, we have

$$|P_j| \leq \max\{\tau - 2 + 2, 2\tau - 5, (2\tau - 5) - 1\} \leq 2\tau - 3$$

for  $j \in \{1, 2, \dots, \tau + 1\}$  with  $\tau \geq 4$ . From these discussions,  $\tau + 1$  paths  $P_1, P_2, \dots, P_{\tau+1}$  joining  $s$  and nodes of  $D$  in  $C_\tau$  are node-disjoint expect for  $s$ , which is consistent with our assumptions.

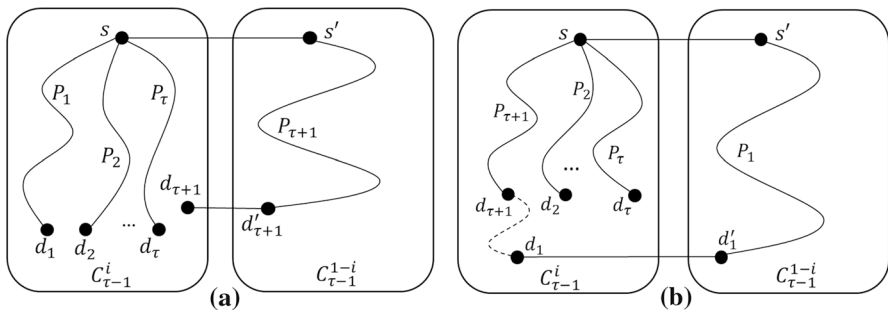


Fig. 5 Illustrations for a Case 1.1 and b Case 1.2 in Lemma 9, respectively

Case 2.  $\alpha = 0$ .

For any integer  $j$  with  $1 \leq j \leq \tau + 1$ , let  $d'_j = (d_j)^\tau$ . Let  $D'' = \{d'_1, d'_2, \dots, d'_\tau\}$ . We further deal with the following two subcases with respect to  $s'$ .

Subcase 2.1.  $s' \in D$ .

Without loss of generality, let  $d_{\tau+1} = s'$ . By the induction hypothesis, there are  $\tau$  node-disjoint paths  $Q_1, Q_2, \dots, Q_\tau$  (except for  $s$ ) joining  $s$  and nodes of  $D''$  in  $C_{\tau-1}^i$  such that  $d'_k \in V(Q_k)$  for  $1 \leq k \leq \tau$ , whose maximum length is limited by  $2\tau - 5$ . For any integer  $1 \leq j \leq \tau + 1$ , if  $1 \leq j \leq \tau$ , let  $P_j = (Q_j, d_j)$ ; otherwise, let  $P_j = (s, d_{\tau+1})$  (refer to Fig. 6a).

By the induction hypothesis and Definition 1, we can verify that the  $\tau$  paths  $P_1, P_2, \dots, P_\tau$  are node-disjoint expect for  $s$ . In addition, we have  $V(P_{\tau+1}) \cap V(P_k) = \{s\}$  for all  $1 \leq k \leq \tau$ . Therefore, we state that  $P_1, P_2, \dots, P_{\tau+1}$  are node-disjoint expect for  $s$ . By the induction hypothesis, we have

$$|P_j| \leq \max\{1, (2\tau - 5) + 1\} = 2\tau - 4 \leq 2\tau - 3$$

for  $j \in \{1, 2, \dots, \tau + 1\}$  with  $\tau \geq 4$ . From these discussions,  $\tau + 1$  paths  $P_1, P_2, \dots, P_{\tau+1}$  joining  $s$  and nodes of  $D$  in  $C_\tau$  are node-disjoint expect for  $s$ , which is consistent with our assumptions.

Subcase 2.2.  $s' \notin D$ .

Let  $D''' = \{d_2, d_3, \dots, d_{\tau+1}\}$ . By the induction hypothesis, there exist  $\tau$  node-disjoint paths  $Q_2, Q_3, \dots, Q_{\tau+1}$  (except for  $s'$ ) joining  $s'$  and nodes of  $D'''$  in  $C_{\tau-1}^{1-i}$  such that  $d_k \in V(Q_k)$  for  $2 \leq k \leq \tau + 1$ , whose maximum length is limited by  $2\tau - 5$ . Choose  $\phi \in \{2, 3, \dots, \tau + 1\}$  with  $d_1 \notin V(Q_\phi)$ . Without loss of generality, let  $\phi = \tau + 1$ . By the induction hypothesis, there exist  $\tau$  node-disjoint paths  $S_1, S_2, \dots, S_\tau$  (except for  $s$ ) joining  $s$  and nodes of  $D''$  in  $C_{\tau-1}^i$  such that  $d_k \in V(S_k)$  for  $1 \leq k \leq \tau$ , whose maximum length is limited by  $2\tau - 5$ . For any integer  $1 \leq j \leq \tau + 1$ , let if  $1 \leq j \leq \tau$ , let  $P_j = (S_j, d_j)$ ; otherwise, let  $P_j = (s, Q_{\tau+1})$  (refer to Fig. 6b).

By the induction hypothesis and Definition 1, we can verify that the  $\tau$  paths  $P_1, P_2, \dots, P_\tau$  are node-disjoint expect for  $s$ . In addition, we have  $V(P_{\tau+1}) \cap V(P_k) = \{s\}$  for all  $1 \leq k \leq \tau$ . Therefore, we state that  $P_1, P_2, \dots, P_{\tau+1}$  are node-disjoint expect for  $s$ . By the induction hypothesis, we have

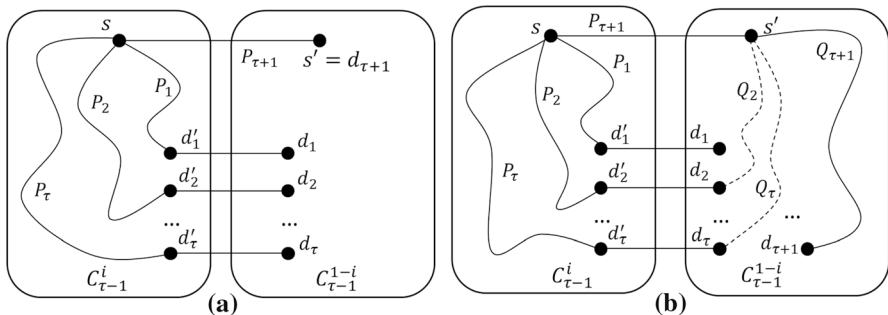


Fig. 6 Illustrations for a Case 2.1 and b Case 2.2 in Lemma 9, respectively

$$|P_j| \leq \max\{(2\tau - 5) + 1, (2\tau - 5) + 1\} = 2\tau - 4 \leq 2\tau - 3$$

for  $j \in \{1, 2, \dots, \tau + 1\}$  with  $\tau \geq 4$ . From these discussions,  $\tau + 1$  paths  $P_1, P_2, \dots, P_{\tau+1}$  joining  $s$  and nodes of  $D$  in  $C_\tau$  are node-disjoint expect for  $s$ , which is consistent with our assumptions.

Case 3.  $1 \leq \alpha \leq \tau$ .

Without loss of generality, denote  $D_1 = \{d_1, d_2, \dots, d_\alpha\}$ ,  $D_2 = \{d_{\alpha+1}, d_{\alpha+2}, \dots, d_{\tau+1}\}$ ,  $D_3 = \{d_1, d_2, \dots, d_\gamma\}$  and  $D_4 = \{d_{\alpha+1}, d_{\alpha+2}, \dots, d_{\alpha+\gamma}\}$  if  $\beta \geq 1$  and  $\gamma \geq 1$ . Clearly, we have  $\gamma \leq \alpha, \gamma \leq \beta$ , and  $\alpha + \beta = \tau + 1$ . For any integer  $j$  with  $\alpha + 1 \leq j \leq \tau + 1$ , let  $d'_j = (d_j)^\tau$ . We further have the following four subcases with respect to  $\gamma$  and  $s'$ .

Subcase 3.1.  $\gamma = 0$  and  $s' \in D$ .

Without loss of generality, let  $s' = d_{\tau+1}$ . Then, let

$$D'' = \begin{cases} \{d_1, d_2, \dots, d_\alpha\} & \text{if } \beta = 1, \\ \{d_1, d_2, \dots, d_\alpha, d'_{\alpha+1}, d'_{\alpha+2}, \dots, d'_\tau\} & \text{if } \beta > 1. \end{cases}$$

By the induction hypothesis, there exist  $\tau$  node-disjoint paths (except for  $s$ )  $Q_1, Q_2, \dots, Q_\tau$  joining  $s$  and nodes of  $D''$  in  $C_{\tau-1}^i$  such that  $d_k \in V(Q_k)$  for all  $1 \leq k \leq \alpha$  and  $d'_k \in V(Q_k)$  for all  $\alpha + 1 \leq k \leq \tau$  if  $\beta > 1$ , whose maximum length is limited by  $2\tau - 5$ . For any integer  $1 \leq j \leq \tau + 1$ , let

$$P_j = \begin{cases} Q_j & \text{if } 1 \leq j \leq \alpha, \\ (Q_j, d_j) & \text{if } \alpha + 1 \leq j \leq \tau \text{ and } \beta > 1, \\ (s, d_{\tau+1}) & \text{if } j = \tau + 1. \end{cases}$$

Accordingly, all the  $\tau + 1$  needed paths connecting  $s$  and nodes of  $D = \{d_1, d_2, \dots, d_{\tau+1}\}$  of  $C_\tau$  are constructed above (refer to Figure 7a, b). By the induction hypothesis and Definition 1, we can verify that the  $\tau$  paths  $P_1, P_2, \dots, P_\tau$  are node-disjoint expect for  $s$ . In addition, we can verify that  $V(P_{\tau+1}) \cap V(P_k) = \{s\}$  for all  $1 \leq k \leq \tau$ . Therefore, we state that  $P_1, P_2, \dots, P_{\tau+1}$  are node-disjoint expect for  $s$ . By Lemma 4, we have

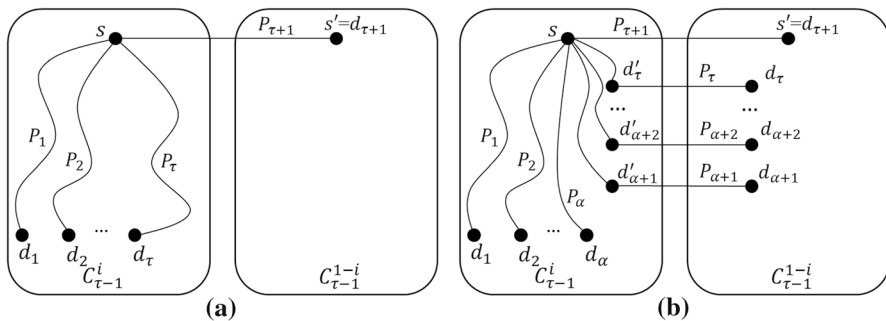


Fig. 7 Illustrations for a  $\beta = 1$  and b  $\beta > 1$  of Case 3.1 in Lemma 9, respectively

$$|P_j| \leq \max\{2\tau - 5, (2\tau - 5) + 1, 1\} \leq 2\tau - 4 \leq 2\tau - 3$$

for  $j \in \{1, 2, \dots, \tau + 1\}$  with  $\tau \geq 4$ . From these discussions,  $\tau + 1$  paths  $P_1, P_2, \dots, P_{\tau+1}$  joining  $s$  and nodes of  $D$  in  $C_\tau$  are node-disjoint expect for  $s$ , which is consistent with our assumptions.

Subcase 3.2.  $\gamma \leq 1$  and  $s' \notin D$ .

Without loss of generality, let  $d_{\tau+1} \in D_4$  if  $\gamma = 1$ . Let

$$D'' = \begin{cases} \{d_1, d_2, \dots, d_\alpha\} & \text{if } \beta = 1, \\ \{d_1, d_2, \dots, d_\alpha, d'_{\alpha+1}, d'_{\alpha+2}, \dots, d'_\tau\} & \text{if } \beta > 1. \end{cases}$$

Then, by the induction hypothesis, there exist  $\tau$  node-disjoint paths (except for  $s$ )  $Q_1, Q_2, \dots, Q_\tau$  joining  $s$  and nodes of  $D''$  in  $C_{\tau-1}^i$  such that  $d_k \in V(Q_k)$  for all  $1 \leq k \leq \alpha$  and  $d'_k \in V(Q_k)$  for all  $\alpha + 1 \leq k \leq \tau$  if  $\beta > 1$ , whose maximum length is limited by  $2\tau - 5$ .

If  $\beta > 1$ . Choose  $\alpha - 1$  distinct nodes (resp. node)  $x_2, x_3, \dots, x_\alpha$  in  $C_{\tau-1}^{1-i}$  such that  $\{x_2, x_3, \dots, x_\alpha\} \cap \{s', d_{\alpha+1}, d_{\alpha+2}, \dots, d_{\tau+1}\} = \emptyset$  when  $\alpha > 1$ . Then, let

$$D''' = \begin{cases} \{d_2, d_3, \dots, d_{\tau+1}\} & \text{if } \alpha = 1, \\ \{x_2, x_3, \dots, x_\alpha, d_{\alpha+1}, d_{\alpha+2}, \dots, d_{\tau+1}\} & \text{if } \alpha > 1. \end{cases}$$

By the induction hypothesis, there exist  $\tau$  node-disjoint paths (except for  $s'$ )  $S_2, S_3, \dots, S_{\tau+1}$  joining  $s'$  and nodes of  $D'''$  in  $C_{\tau-1}^{1-i}$  such that  $x_k \in V(S_k)$  for  $2 \leq k \leq \alpha$  if  $\alpha > 1$  and  $d_k \in V(S_k)$  for  $\alpha + 1 \leq k \leq \tau + 1$ , whose maximum length is limited by  $2\tau - 5$ .

For any integer  $1 \leq j \leq \tau + 1$ , let

$$P_j = \begin{cases} Q_j & \text{if } 1 \leq j \leq \alpha, \\ (Q_j, d_j) & \text{if } \alpha + 1 \leq j \leq \tau \text{ and } \beta > 1, \\ (s, S_{\tau+1}) & \text{if } j = \tau + 1 \text{ and } \beta > 1, \\ (s, \text{CRouting}(C_{\tau-1}^{1-i}, s', d_{\tau+1})) & \text{if } j = \tau + 1 \text{ and } \beta = 1. \end{cases}$$

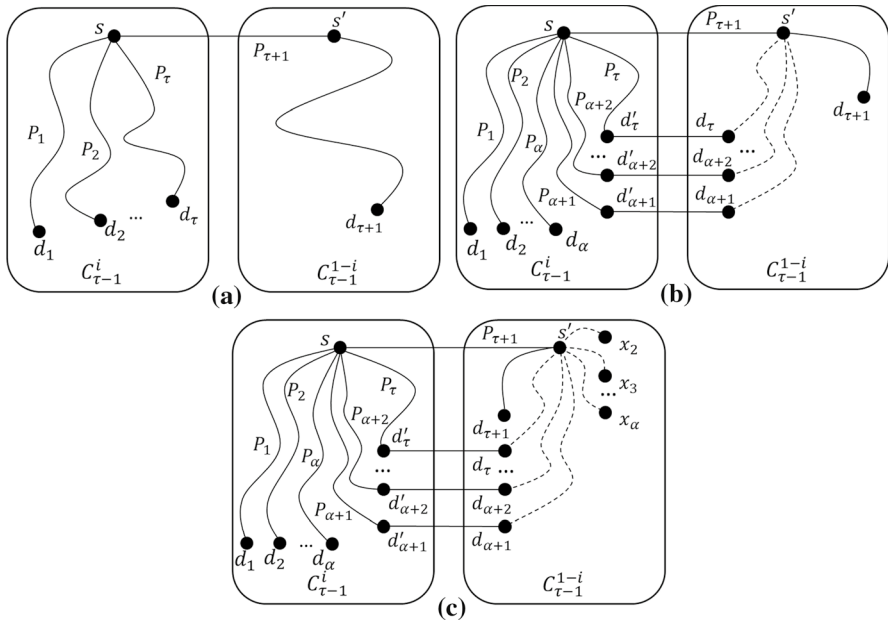
Furthermore, all the  $\tau + 1$  needed paths connecting  $s$  and nodes of  $D = \{d_1, d_2, \dots, d_{\tau+1}\}$  of  $C_\tau$  are constructed above (refer to Fig. 8a–c). By the induction hypothesis and Definition 1, we can verify that the  $\tau$  paths  $P_1, P_2, \dots, P_\tau$  are node-disjoint expect for  $s$ . In addition, by the induction hypothesis, we can verify that  $V(P_{\tau+1}) \cap V(P_k) = \{s\}$  for all  $1 \leq k \leq \tau$ . Therefore, we state that  $P_1, P_2, \dots, P_{\tau+1}$  are node-disjoint expect for  $s$ . By Lemma 4, we have

$$|P_j| \leq \max\{2\tau - 5, (2\tau - 5) + 1, (2\tau - 5) + 1, \tau - 2 + 1\} \leq 2\tau - 4 < 2\tau - 3$$

for  $j \in \{1, 2, \dots, \tau + 1\}$  with  $\tau \geq 4$ . From these discussions,  $\tau + 1$  paths  $P_1, P_2, \dots, P_{\tau+1}$  joining  $s$  and nodes of  $D$  in  $C_\tau$  are node-disjoint expect for  $s$ , which is consistent with our assumptions.

Subcase 3.3.  $\gamma \geq 1$  and  $s' \in D$ .

Without loss of generality, let  $s' = d_{\tau+1}$  and  $d_j = (d_{\alpha+j})^\gamma$  for  $j \in \{1, 2, \dots, \gamma\}$ . For any integer  $j$  with  $\gamma + 1 \leq j \leq \alpha$  and  $\alpha + \gamma + 1 \leq j \leq \tau$ , let  $d'_j = (d_j)^\gamma$ . Denote



**Fig. 8** Illustrations for **a**  $\beta = 1$ , **b**  $\beta > 1$  and  $\alpha = 1$ , and **c**  $\beta > 1$  and  $\alpha > 1$  of Case 3.2 in Lemma 9, respectively

$$T = \begin{cases} \{d_1, d_2, \dots, d_\alpha, s, s', d'_{\gamma+1}, d'_{\gamma+2}, \dots, d'_\alpha\} & \text{if } \gamma = \beta - 1, \\ \{d_1, d_2, \dots, d_\alpha, s, s', d'_{\gamma+1}, d'_{\gamma+2}, \dots, d'_\alpha, d_{\alpha+\gamma+1}\}, & \\ \{d_{\alpha+\gamma+2}, \dots, d_\tau, d'_{\alpha+\gamma+1}, d'_{\alpha+\gamma+2}, \dots, d'_\tau\} & \text{if } \gamma < \beta - 1. \end{cases}$$

and

$$k = \begin{cases} \alpha - \gamma + 1 & \text{if } \gamma = \beta - 1, \\ \alpha - \gamma + 1 + \tau - (\alpha + \gamma) & \text{if } \gamma < \beta - 1. \end{cases}$$

Since  $\alpha + \beta = \tau + 1$ ,  $\beta \geq \gamma + 1$ , and  $\alpha \geq \gamma$ , we have  $\gamma \leq \frac{\tau}{2} = \lceil \frac{\tau-1}{2} \rceil$  and

$$k = \begin{cases} \tau + 1 - \beta - \gamma + 1 = \tau - 2\gamma + 1 & \text{if } \gamma = \beta - 1, \\ \alpha - \gamma + 1 + \tau - (\alpha + \gamma) = \tau - 2\gamma + 1 & \text{if } \gamma < \beta - 1. \end{cases}$$

If  $\gamma = 1$ , by Lemma 8, there exists a path  $S_{\alpha+1} = (d_{\alpha+1}, x_{\alpha+1}, d'_{\alpha+1})$  such that  $(d_{\alpha+1}, x_{\alpha+1}) \in E(C_{\tau-1}^{1-i})$ ,  $d'_{\alpha+1} = (x_{\alpha+1})^\tau$ , and  $V(S_{\alpha+1}) \cap T = \emptyset$ ; If  $\gamma > 1$ , by Lemma 8, there exist  $\gamma$  node-disjoint paths  $S_{\alpha+1} = (d_{\alpha+1}, x_{\alpha+1}, d'_{\alpha+1})$ ,  $S_{\alpha+2} = (d_{\alpha+2}, x_{\alpha+2}, d'_{\alpha+2})$ ,  $\dots$ ,  $S_{\alpha+\gamma} = (d_{\alpha+\gamma}, x_{\alpha+\gamma}, d'_{\alpha+\gamma})$  such that  $(d_j, x_j) \in E(C_{\tau-1}^{1-i})$ ,  $d'_j = (x_j)^\tau$ , and  $V(S_j) \cap T = \emptyset$  for  $j \in \{\alpha + 1, \alpha + 2, \dots, \alpha + \gamma\}$ .

Furthermore, let  $D'' = \{d_1, d_2, \dots, d_\alpha, d'_{\alpha+1}, d'_{\alpha+2}, \dots, d'_\tau\}$ . Hence, we can verify that  $D'' \cap \{s\} = \emptyset$  and  $|D''| = \tau$ . By the induction hypothesis, there exist  $\tau$  node-disjoint paths (except for  $s$ )  $Q_1, Q_2, \dots, Q_\tau$  joining  $s$  and nodes of  $D''$  in  $C_{\tau-1}^i$  such

that  $d_k \in V(Q_k)$  for  $1 \leq k \leq \alpha$  and  $d'_k \in V(Q_k)$  for  $\alpha + 1 \leq k \leq \tau$ , whose maximum length is limited by  $2\tau - 5$ . For any integer  $1 \leq j \leq \tau + 1$ , let

$$P_j = \begin{cases} Q_j & \text{if } 1 \leq j \leq \alpha, \\ (Q_j, x_j, d_j) & \text{if } \alpha + 1 \leq j \leq \alpha + \gamma, \\ (Q_j, d_j) & \text{if } \alpha + \gamma + 1 \leq j \leq \tau \text{ and } \gamma < \beta - 1, \\ (s, d_{\tau+1}) & \text{if } j = \tau + 1. \end{cases}$$

Accordingly, all the  $\tau + 1$  needed paths connecting  $s$  and nodes of  $D = \{d_1, d_2, \dots, d_{\tau+1}\}$  of  $C_\tau$  are constructed above (refer to Fig. 9a, b). By the induction hypothesis, Lemma 8, and Definition 1, we can verify that the  $\tau$  paths  $P_1, P_2, \dots, P_\tau$  are node-disjoint expect for  $s$ . In addition, by the induction hypothesis, we can verify that  $V(P_{\tau+1}) \cap V(P_k) = \{s\}$  for all  $1 \leq k \leq \tau$ . Therefore, we state that  $P_1, P_2, \dots, P_{\tau+1}$  are node-disjoint expect for  $s$ . By Lemma 4, we have

$$|P_j| \leq \max\{2\tau - 5, (2\tau - 5) + 1 + 1, (2\tau - 5) + 1, 1\} \leq 2\tau - 3$$

for  $j \in \{1, 2, \dots, \tau + 1\}$  with  $\tau \geq 4$ . From these discussions,  $\tau + 1$  paths  $P_1, P_2, \dots, P_{\tau+1}$  joining  $s$  and nodes of  $D$  in  $C_\tau$  are node-disjoint expect for  $s$ , which is consistent with our assumptions.

Subcase 3.4.  $\gamma \geq 2$  and  $s' \notin D$ .

Without loss of generality, let  $d_j = (d_{\alpha+j})^\tau$  for  $j \in \{1, 2, \dots, \gamma\}$ . For any integer  $j$  with  $\gamma + 1 \leq j \leq \alpha$  and  $\alpha + \gamma + 1 \leq j \leq \tau + 1$ , let  $d'_j = (d_j)^\tau$ . Denote  $\phi = \gamma - 1$ ,  $k = 2 + (\beta - \gamma)$ , and

$$T = \begin{cases} \{d_1, d_2, \dots, d_\alpha, s, s', d_{\tau+1}\} & \text{if } \gamma = \beta, \\ \{d_1, d_2, \dots, d_\alpha, s, s', d_{\alpha+\gamma}, d_{\alpha+\gamma+1}, \dots, d_{\tau+1}, d'_{\alpha+\gamma+1}, d'_{\alpha+\gamma+2}, \dots, d'_{\tau+1}\} & \text{if } \gamma < \beta. \end{cases}$$

Since  $\alpha + \beta = \tau + 1$ ,  $\gamma \leq \beta$ , and  $\alpha \geq \gamma$ , we have

$$1 \leq \phi = \gamma - 1 \leq \frac{\tau + 1}{2} - 1 \leq \lceil \frac{\tau - 1}{2} \rceil$$

and

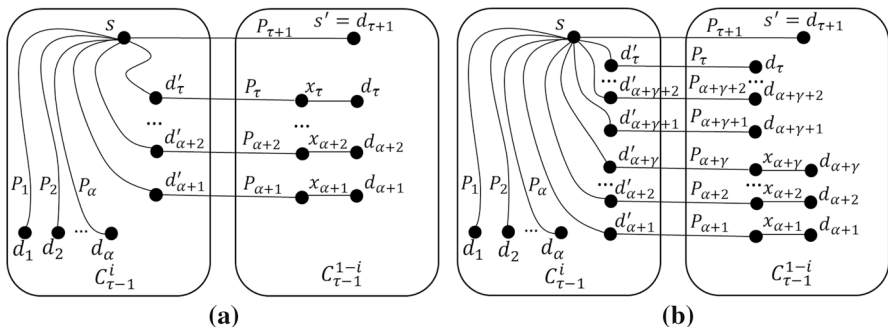


Fig. 9 Illustrations for a  $\gamma = \beta - 1$  and b  $\gamma < \beta - 1$  of Case 3.3 in Lemma 9, respectively

$$k = 2 + (\beta - \gamma) = \tau + 1 - (\alpha + \gamma) + 2 \leq \tau - 2(\gamma - 1) + 1 = \tau - 2\phi + 1.$$

If  $\phi = 1$ , by Lemma 8, there exists a path  $Q'_{\alpha+1} = (d_{\alpha+1}, x_{\alpha+1}, d'_{\alpha+1})$  such that  $(d_{\alpha+1}, x_{\alpha+1}) \in E(C_{\tau-1}^{1-i})$ ,  $d'_{\alpha+1} = (x_{\alpha+1})^\tau$ , and  $V(Q'_{\alpha+1}) \cap T = \emptyset$ ; If  $\phi > 1$ , by Lemma 8, there exist  $\phi$  node-disjoint paths  $Q'_{\alpha+1} = (d_{\alpha+1}, x_{\alpha+1}, d'_{\alpha+1})$ ,  $Q'_{\alpha+2} = (d_{\alpha+2}, x_{\alpha+2}, d'_{\alpha+2})$ ,  $\dots$ ,  $Q'_{\alpha+\phi} = (d_{\alpha+\phi}, x_{\alpha+\phi}, d'_{\alpha+\phi})$  such that  $(d_j, x_j) \in E(C_{\tau-1}^{1-i})$ ,  $d'_j = (x_j)^\tau$ , and  $V(Q'_j) \cap T = \emptyset$  for  $j \in \{\alpha + 1, \alpha + 2, \dots, \alpha + \phi\}$ .

Let  $\varphi = \tau - 2\phi - (\beta - \phi)$ . Accordingly, we have

$$\varphi = \tau - 2\phi - (\beta - \phi) = \alpha + \beta - 1 - \phi - \beta = \alpha - (\phi + 1) = \alpha - \gamma$$

and thus we have  $0 \leq \varphi < \alpha$ . If  $\varphi \geq 0$ , we have  $\alpha = \gamma$ . If  $\varphi \geq 1$ , choose  $\varphi$  distinct nodes (resp. node)  $x_1, x_2, \dots, x_\varphi$  in  $C_{\tau-1}^{1-i}$  such that  $\{x_1, x_2, \dots, x_\varphi\} \cap \{s', d_{\alpha+1}, d_{\alpha+2}, \dots, d_{\tau+1}, x_{\alpha+1}, x_{\alpha+2}, \dots, x_{\alpha+\phi}\} = \emptyset$ . Furthermore, let

$$D'' = \begin{cases} \{d_{\alpha+1}, d_{\alpha+2}, \dots, d_{\tau+1}, x_{\alpha+1}, x_{\alpha+2}, \dots, x_{\alpha+\phi}\} & \text{if } \varphi = 0, \\ \{d_{\alpha+1}, d_{\alpha+2}, \dots, d_{\tau+1}, x_1, x_2, \dots, x_\varphi, x_{\alpha+1}, x_{\alpha+2}, \dots, x_{\alpha+\phi}\} & \text{if } \varphi \geq 1. \end{cases}$$

and  $D''' = \{d_1, d_2, \dots, d_\alpha, d'_{\alpha+1}, d'_{\alpha+2}, \dots, d'_{\alpha+\phi}, d'_{\alpha+\gamma+1}, d'_{\alpha+\gamma+2}, \dots, d'_{\tau+1}\}$ . Hence, we can verify that  $D'' \cap \{s'\} = \emptyset$ ,  $D''' \cap \{s\} = \emptyset$ ,  $|D''| = \beta + \alpha - 1 = \tau$ , and

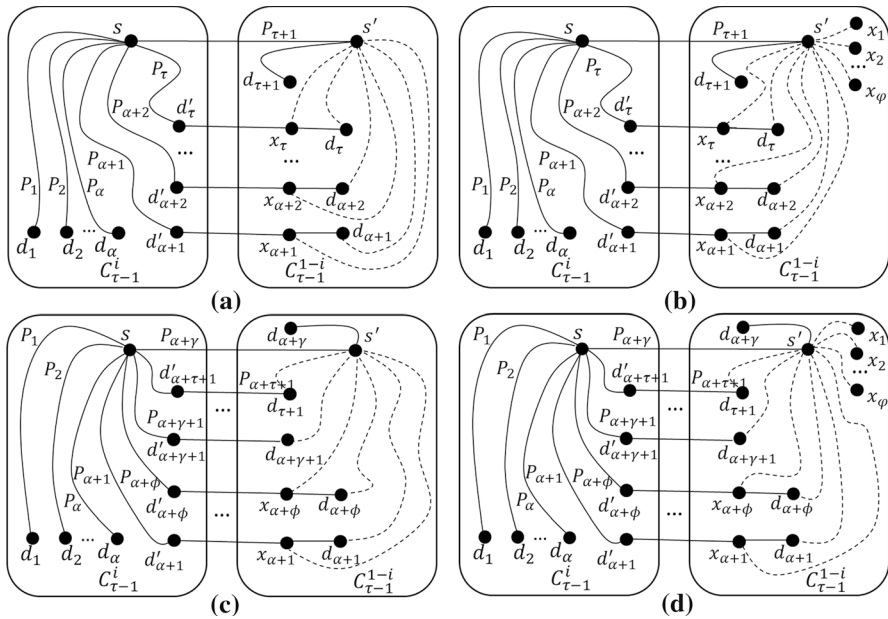
$$|D''| = \begin{cases} \beta + \phi = \beta + \alpha - 1 = \tau & \text{if } \varphi = 0, \\ \beta + \phi + \varphi = \beta + \gamma - 1 + \alpha - \gamma = \beta + \alpha - 1 = \tau & \text{if } \varphi \geq 1. \end{cases}$$

By the induction hypothesis, there exist  $\tau$  node-disjoint paths (except for  $s'$ )  $Q_2, Q_3, \dots, Q_{\tau+1}$  joining  $s'$  and nodes of  $D''$  in  $C_{\tau-1}^{1-i}$  such that  $d_k \in V(Q_k)$  for all  $\alpha + 1 \leq k \leq \tau + 1$ ,  $x_{k+\phi} \in V(Q_k)$  for all  $\alpha - \phi + 1 \leq k \leq \alpha$ , and  $x_{k-1} \in V(Q_k)$  for all  $2 \leq k \leq \alpha - \phi$  if  $\varphi \geq 1$ , whose maximum length is limited by  $2\tau - 5$ . Moreover, by the induction hypothesis, there exist  $\tau$  node-disjoint paths (except for  $s$ )  $S_1, S_2, \dots, S_{\alpha+\phi}, S_{\alpha+\gamma+1}, S_{\alpha+\gamma+2}, \dots, S_{\tau+1}$  joining  $s$  and nodes of  $D'''$  in  $C_{\tau-1}^i$  such that  $d_k \in V(S_k)$  for  $1 \leq k \leq \alpha$  and  $d'_k \in V(S_k)$  for  $\alpha + 1 \leq k \leq \alpha + \phi$  and  $\alpha + \gamma + 1 \leq k \leq \tau + 1$ , whose maximum length is limited by  $2\tau - 5$ . For any integer  $1 \leq j \leq \tau + 1$ , let

$$P_j = \begin{cases} S_j & \text{if } 1 \leq j \leq \alpha, \\ (S_j, x_j, d_j) & \text{if } \alpha + 1 \leq j \leq \alpha + \phi, \\ (s, Q_j) & \text{if } j = \alpha + \gamma, \\ (S_j, d_j) & \text{if } \alpha + \gamma + 1 \leq j \leq \tau + 1. \end{cases}$$

Accordingly, all the  $\tau + 1$  needed paths connecting  $s$  and nodes of  $D = \{d_1, d_2, \dots, d_{\tau+1}\}$  of  $C_\tau$  are constructed above (refer to Fig. 10a–d). By the induction hypothesis, Lemma 8, and Definition 1, we can verify that the  $\tau$  paths  $P_1, P_2, \dots, P_\tau$  are node-disjoint expect for  $s$ . In addition, by the induction hypothesis, we can verify that  $V(P_{\tau+1}) \cap V(P_k) = \{s\}$  for all  $1 \leq k \leq \tau$ . Therefore, we state that  $P_1, P_2, \dots, P_{\tau+1}$  are node-disjoint expect for  $s$ . By Lemma 4, we have

$$|P_j| \leq \max\{2\tau - 5, (2\tau - 5) + 1 + 1, (2\tau - 5) + 1, (2\tau - 5) + 1\} \leq 2\tau - 3$$



**Fig. 10** Illustrations for **a**  $\gamma = \beta$  and  $\varphi = 0$ , **b**  $\gamma = \beta$  and  $\varphi \geq 1$ , **c**  $\gamma \leq \beta - 1$  and  $\varphi = 0$ , and **d**  $\gamma \leq \beta - 1$  and  $\varphi \geq 1$  of Case 3.4 in Lemma 9, respectively

for  $j \in \{1, 2, \dots, \tau + 1\}$  with  $\tau \geq 4$ . From these discussions,  $\tau + 1$  paths  $P_1, P_2, \dots, P_{\tau+1}$  joining  $s$  and nodes of  $D$  in  $C_\tau$  are node-disjoint except for  $s$ , which is consistent with our assumptions.

In summary, the lemma holds for  $n = \tau$ . □

By Lemma 5 and 9, the following theorem holds.

**Theorem 10** For any integer  $n \geq 2$ , choose an arbitrary source node  $s$  and an arbitrary node set  $D = \{d_1, d_2, \dots, d_{n+1}\}$  in a  $C_n$  with  $s \notin D$ , there exist  $n + 1$  node-disjoint paths (except for  $s$ ) from  $s$  to nodes of  $D$ , whose maximum length is limited by  $2n - 3$ .

Consider two nodes  $s = 00000$  and  $D = \{01001, 01010, 01111, 11001, 11010, 11111\}$  in  $C_5$ . The construction of the six node-disjoint paths falls into Lemma 9. The following six paths (refer to Figure 11) are constructed according to Lemma 9, respectively.

- $P_1 = (00000, 00001, 01001),$
- $P_2 = (00000, 00010, 01010),$
- $P_3 = (00000, 01000, 01011, 01111),$
- $P_4 = (00000, 10000, 11000, 11001),$
- $P_5 = (00000, 00011, 00111, 00110, 01110, 11110, 11010),$  and
- $P_6 = (00000, 00100, 01100, 11100, 11111).$



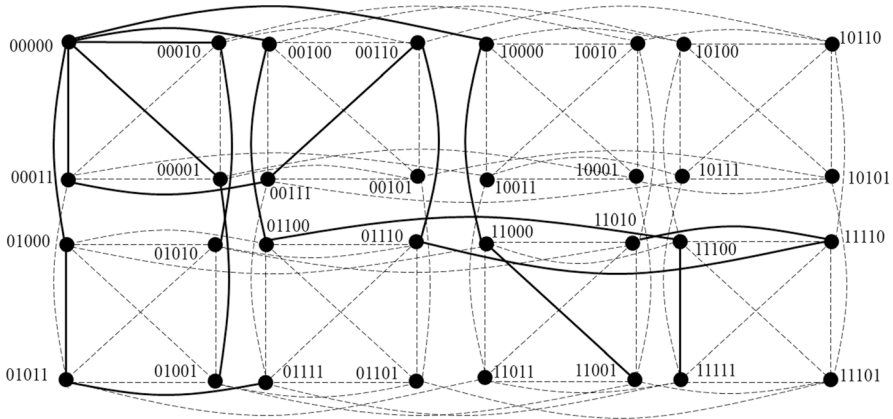


Fig. 11 Six node-disjoint paths from  $s = 00000$  to nodes of  $D = \{01001, 01010, 01111, 11001, 11010, 11111\}$  in  $C_5$  are constructed according to Theorem 10

The maximum length of the six node-disjoint paths is computed as 6. Actually, Theorem 10 took with the worst-case consideration of the maximal length of  $n + 1$  node-disjoint paths between node  $s$  and nodes of  $D$  in  $C_n$ .

### 3.2 A Constructive Node-to-set Algorithm of Cross-cubes

In this subsection, we give an efficient algorithm, Algorithm N2SMain, to find  $n + 1$  node-to-set disjoint paths in  $C_n$  if  $n \geq 2$ . Then, we analyze the time complexity of the Algorithm N2SMain.

**Theorem 11** For any integer  $n \geq 2$ , given any arbitrary source node  $s$  and an arbitrary node set  $D = \{d_1, d_2, \dots, d_{n+1}\}$  in a  $C_n$  with  $s \notin D$ , there exist an  $O(N \log^2 N)$  algorithm for finding  $n + 1$  node-disjoint paths (except for  $s$ ) from  $s$  to nodes of  $D$  of  $C_n$ , where  $N$  is the node number of  $C_n$ .

**Proof** Considering node-to-set disjoint paths for the given node  $u$  and set  $D$  in  $C_n$  with  $s \notin D$ , we propose an efficient algorithm, Algorithm N2SMain. To simplify the presentation of the proposed routing algorithm, we first introduce two algorithms, namely CProute and CSet, that will be the two core components of the proposed algorithm.

In order to compactly express the algorithm, we need some notations. We use a dictionary as a data structure that represents a collection of keys and values pair of data. Let  $T$  be a dictionary, we use  $T = [ ]$  to denote a empty dictionary which contains zero element. Then, let  $a, b \in V(G)$  be two nodes of  $G$ , we use  $T(a) = b$  to set  $b$  as a value of key  $a$  and  $T(a)$  to return the item of  $b$  with key  $a$  in  $T$ , respectively. Let  $S$  be a set with  $m$  elements, we use  $|S|$  to denote the number of elements in a set  $S$  such that  $|S| = m$ . Furthermore, we use  $S.add(a)$  to add element  $a$  to the set  $S$ ,  $S.remove(a)$  to remove element  $a$  from the set  $S$ , and  $S = \emptyset$  to denote a empty set with  $|S| = 0$ , respectively. □

**Algorithm 1** CProute

**Input:** A set of  $n$  node-disjoint paths  $\{Q_1, Q_2, \dots, Q_n\}$ , two node sets  $D$  and  $D'$ , and a dictionary of nodes  $T$  of  $C_n$ .

```

1: function CROUTE1( $\{Q_1, Q_2, \dots, Q_n\}, D, D', T$ )
2:   for  $i \in \{1, 2, \dots, n\}$  do
3:     Let  $x \leftarrow Q_i[-1]$ ;
4:     if  $x \in D$  then
5:        $P_i \leftarrow (Q_i, (x)^n)$ ;
6:     else if  $x \in D'$  then
7:        $P_i \leftarrow (Q_i, (x)^n, T[x])$ ;
8:     else
9:        $P_i \leftarrow Q_i$ ;
10:    end if
11:  end for
12:  return  $\{P_1, P_2, \dots, P_n\}$ ;
13: end function
14: function CROUTE2( $\{Q_1, Q_2, \dots, Q_n\}, D$ )
15:   for  $i \in \{1, 2, \dots, n\}$  do
16:     if  $D \cap V(Q_i) = \emptyset$  then
17:       return  $Q_i$ ;
18:     end if
19:   end for
20: end function
21: function CROUTE3( $\{Q_1, Q_2, \dots, Q_n\}, D$ )
22:   for  $i \in \{1, 2, \dots, n\}$  do
23:     if  $D \cap V(Q_i) \neq \emptyset$  then
24:       return  $Q_i$ ;
25:     end if
26:   end for
27: end function

```

We first study the Algorithm CProute. In line 3, it takes  $O(1)$  time to get the last node from the path  $Q_i$  with  $i \in \{1, 2, \dots, n\}$ . In lines 5 and 7, we assume, in our time analysis, that  $(x)^n$  can be computed in constant time, which is the case when using the connection rule given in Definition 1. In line 7, it takes  $O(1)$  time to compute  $T[x]$ . In line 5, it takes constant time to construct  $P_i$  by joining path  $Q_i$  and node  $(x)^n$ . In line 7, it takes  $O(1)$  time to construct  $P_i$  by joining path  $Q_i$  and nodes  $(x)^n, T[x]$ . In line 12, it takes constant time to return  $n$  paths  $\{P_1, P_2, \dots, P_n\}$ . Accordingly, we can verify that the time complexity of function CRoute1 in Algorithm CProute is  $O(n)$ . In lines 16 and 23, it takes  $O(1)$  time to compute  $V(Q_i)$ . In lines 16 and 23, it takes  $O(n)$  time to compute  $D \cap V(Q_i)$ . In lines 17 and 24, it takes constant time to return a path  $Q_i$  with  $i \in \{1, 2, \dots, n\}$ . Thus, we can verify that the time complexity of function CRoute2 and CRoute3 in Algorithm CProute is  $O(n^2)$ .

**Algorithm 2** CSet

**Input:** Two node sets  $D, M$  and a subgraph  $G$  of  $C_n$ , a binary string  $j$ , and an integer  $\phi$ .

```

1: function CSet1( $D, M, G, j$ )
2:   Let  $H \leftarrow \emptyset, S \leftarrow \emptyset$ , and  $T \leftarrow []$ ;
3:   for  $u \in D$  do
4:     for  $v \in N_G(u)$  do
5:       if  $v \in M$  then
6:         Let  $x \leftarrow (v)^n, H.add(x), S.add(v), T[v] \leftarrow u$ , and  $M.remove(v)$ ;
7:         break;
8:       end if
9:     end for
10:  end for
11:  if  $j = 0$  then
12:    return  $(H, T)$ ;
13:  else
14:    return  $(H, S, T, M)$ ;
15:  end if
16: end function
17: function CSet2( $D, \phi$ )
18:  Choose  $\phi$  distinct nodes  $x_1, x_2, \dots, x_\phi \in D$ ;
19:  return  $\{x_1, x_2, \dots, x_\phi\}$ ;
20: end function

```

In addition, we propose Algorithm CSet. In lines 4 and 6, we assume, in our time analysis, that  $(x)^n$  and  $N_G(u)$  can be computed in constant time, which is the case when using the connection rule given in Definition 1. In lines 2 and 6 of Algorithm CSet, it takes constant time to compute  $H$  (resp.  $S, T, H.add(x), S.add(v), T[v]$ , and  $M.remove(v)$ ). Accordingly, we can verify that the time complexity of function CSet1 in Algorithm CSet is  $O(n^2)$ . In line 18, it takes  $O(n)$  time to choose  $\phi$  distinct nodes  $x_1, x_2, \dots, x_\phi$  from a node set  $D$ . Thus, we can verify that the time complexity of function CSet2 in Algorithm CSet is  $O(n)$ .

**Algorithm 3**  $N2SMain(s, D, C_n)$ **Input:** A node  $s$  and a node set  $D = \{d_1, d_2, \dots, d_{n+1}\}$  in  $C_n$  where  $s \notin D$ .**Output:**  $n + 1$  disjoint paths except for  $s$  in  $C_n$  joining  $s$  and  $D$ .

```

1: if  $n = 2$  then
2:   return  $\{(s, d) | d \in D\}$ ; ▷ Lemma 5.
3: end if
4: Let  $i \leftarrow \lfloor n/2 \rfloor$ ,  $D_1 \leftarrow D \cap V(C_{n-1}^i)$ ,  $D_2 \leftarrow D \setminus D_1$ ,  $D_3 \leftarrow \{u | u \in D_1 \text{ and } (u)^n \in D\}$ , and  $D_4 \leftarrow \{(u)^n | u \in D_3\}$ ;
5: Let  $\alpha \leftarrow |D_1|$ ,  $\beta \leftarrow |D_2|$ ,  $\gamma \leftarrow |D_3|$ , and  $s' = (s)^n$ ;
6: if  $\alpha = n + 1$  then ▷ Case 1 in Lemma 9.
7:   Let  $\{P_1, P_2, \dots, P_n\} \leftarrow N2SMain(s, \{d_1, d_2, \dots, d_n\}, C_{n-1}^i)$ ;
8:   if  $d_{n+1} \notin \bigcup_{j=1}^n V(P_j)$  then
9:     Let  $P_{n+1} \leftarrow (s, CRouting(C_{n-1}^{1-i}, s', (d_{n+1})^n), d_{n+1})$ ;
10:  else
11:    Choose  $\phi$  such that  $d_\phi, d_{n+1} \in V(P_\phi)$  and  $1 \leq \phi \leq n$ ;
12:    Let  $P_\phi \leftarrow Path(P_{n+1}, s, d_{n+1})$  and  $P_{n+1} \leftarrow (s, CRouting(C_{n-1}^{1-i}, s', (d_\phi)^n), d_\phi)$ ;
13:  end if
14: else if  $\alpha = 0$  then ▷ Case 2 in Lemma 9.
15:  if  $s' \in D$  then
16:    Let  $P_{n+1} \leftarrow (s, s')$ ,  $D' \leftarrow D \setminus \{s'\}$ , and  $D_5 \leftarrow \{(d)^n | d \in D'\}$ ;
17:    Let  $\{P_1, P_2, \dots, P_n\} \leftarrow CProute1(N2SMain(s, D_5, C_{n-1}^i), D_5, \emptyset, \emptyset)$ ;
18:  else
19:    Let  $D' \leftarrow D \setminus \{d_1\}$ ,  $Q \leftarrow CProute2(N2SMain(s', D', C_{n-1}^{1-i}), \{d_1\})$  and  $D_5 \leftarrow \{(d)^n | d \in D \setminus \{Q[-1]\}\}$ ;
20:    Let  $\{P_1, P_2, \dots, P_n\} \leftarrow CProute1(N2SMain(s, D_5, C_{n-1}^i), D_5, \emptyset, \emptyset)$  and  $P_{n+1} \leftarrow (s, Q)$ ;
21:  end if
22: else if  $1 \leq \alpha \leq \tau$  then ▷ Case 3 in Lemma 9.
23:   Let  $d_j^n = (d_j)^n$  such that  $d_j \in D_2$  and  $1 \leq j \leq n + 1$ ;
24:   if  $\gamma = 0$  and  $s' \in D$  then
25:     if  $\beta = 1$  then
26:       Let  $\{P_1, P_2, \dots, P_n\} \leftarrow N2SMain(s, D_1, C_{n-1}^i)$  and  $P_{n+1} \leftarrow (s, s')$ ;
27:     else
28:       Let  $P_{n+1} \leftarrow (s, s')$  and  $D_5 \leftarrow \{(x)^n | x \in D_2 \setminus \{s'\}\}$ ;
29:       Let  $\{P_1, P_2, \dots, P_n\} \leftarrow CProute1(N2SMain(s, D_1 \cup D_5, C_{n-1}^i), D_5, \emptyset, \emptyset)$ ;
30:     end if
31:   else if  $\gamma \leq 1$  and  $s' \notin D$  then
32:     if  $\beta = 1$  then
33:       Let  $D_2 = \{x\}$ ,  $\{P_1, P_2, \dots, P_n\} \leftarrow N2SMain(s, D_1, C_{n-1}^i)$ , and  $P_{n+1} \leftarrow (s, CRouting(C_{n-1}^{1-i}, s', x))$ ;
34:     else
35:       Let  $D' \leftarrow V(C_{n-1}^{1-i}) \setminus (D_2 \cup \{s'\})$  and  $\{Q_1, Q_2, \dots, Q_n\} \leftarrow N2SMain(s', D_2 \cup CSet2(D', \alpha - 1), C_{n-1}^{1-i})$ ;
36:       if  $\gamma = 0$  then
37:         Let  $Q \leftarrow CProute3(\{Q_1, Q_2, \dots, Q_n\}, D_2)$ ;
38:       else
39:         Let  $Q \leftarrow CProute3(\{Q_1, Q_2, \dots, Q_n\}, D_4)$ ;
40:       end if
41:       Let  $D_5 \leftarrow \{(x)^n | x \in D_2 \setminus \{Q[-1]\}\}$ ;
42:       Let  $P_{n+1} \leftarrow (s, Q)$  and  $\{P_1, P_2, \dots, P_n\} \leftarrow CProute1(N2SMain(s, D_1 \cup D_5, C_{n-1}^i), D_5, \emptyset, \emptyset)$ ;
43:     end if
44:   else if  $\gamma \geq 1$  and  $s' \in D$  then
45:     Let  $D_5 \leftarrow \{(x)^n | x \in D_2 \setminus (D_4 \cup \{s'\})\}$  and  $D' \leftarrow V(C_{n-1}^{1-i}) \setminus (D_2 \cup \{(x)^n | x \in D_1\})$ ;
46:     Let  $D_6, T \leftarrow CSet1(D_4, D', C_{n-1}^{1-i}, 0)$ ;
47:     Let  $P_{n+1} \leftarrow (s, s')$  and  $\{P_1, P_2, \dots, P_n\} \leftarrow CProute1(N2SMain(s, D_1 \cup D_5 \cup D_6, C_{n-1}^i), D_5, D_6, T)$ ;
48:   else if  $\gamma > 1$  and  $s' \notin D$  then
49:     Let  $D_5 \leftarrow \{(x)^n | x \in D_2 \setminus D_4 \cup \{s'\}\}$  and  $D' \leftarrow V(C_{n-1}^{1-i}) \setminus (D_2 \cup \{s'\} \cup \{(x)^n | x \in D_1\})$ ;
50:     Let  $D_6, D'_6, T, M \leftarrow CSet1(D_4, D', C_{n-1}^{1-i}, 1)$  and  $D'' \leftarrow V(C_{n-1}^{1-i}) \setminus M$ ;
51:     Let  $Q \leftarrow CProute3(N2SMain(s', D_2 \cup D'_6 \cup CSet2(D'', \alpha - \gamma), C_{n-1}^{1-i}), D_4)$ ;
52:     Let  $P_{n+1} \leftarrow (s, Q)$  and  $\{P_1, P_2, \dots, P_n\} \leftarrow CProute1(N2SMain(s, D_1 \cup D_5 \cup D_6, C_{n-1}^i), D_5, D_6, T)$ ;
53:   end if
54: end if
55: return  $\{P_1, P_2, \dots, P_{n+1}\}$ ;

```

Accordingly, we propose our main algorithm, Algorithm  $N2SMain$ . Given a node  $s$  and a node set  $D = \{d_1, d_2, \dots, d_{n+1}\}$  in  $C_n$  where  $s \notin D$ , we construct  $n + 1$  node-disjoint paths except for  $s$  in  $C_n$  joining  $s$  and  $D$ . Suppose that a path in algorithm  $N2SMain$  is saved by a doubly linked circular list whose head  $u$  and tail  $v$  are pointed by two pointers. Furthermore, each node is stored by a tuple.

In what follows, we will analyze the time complexity of Algorithm N2SMain as follows. In lines 1 ~ 3 of Algorithm N2SMain, it takes constant time to construct the required node-disjoint paths. In lines 4 ~ 5 of Algorithm N2SMain, it takes  $O(1)$  time to compute  $i$  (resp.  $\alpha$ ,  $\beta$ ,  $\gamma$ , and  $s'$ ). In lines 9, 12, and 33 of Algorithm N2SMain, it takes  $O(n)$  time to construct the required path from  $u$  to  $v$  using the  $\text{CRouting}(C_n, u, v)$  function in  $C_n$  [20]. In lines 4, 11, 16, 19, 28, 35, 41, 45, and 49 of Algorithm N2SMain, it takes  $O(n)$  time to compute  $D_1$  (resp.  $D_2$ ,  $D_3$ ,  $D_4$ ,  $\phi$ ,  $D_5$ , and  $D'$ ).

We use  $U(s, D, n)$  to denote the time of finding  $n + 1$  node-to-set disjoint paths between  $s$  and  $D$  in  $C_n$ . Furthermore, we assume that  $n$  is sufficiently large. Let

$$T(n) = \max\{U(s, D, n) \mid \{s\}, D \subset V(C_n) \text{ and } s \notin D\}. \quad (3.1)$$

Accordingly, we have  $T(2) = O(1)$ . We can claim the following discussions with respect to  $n$  for  $n \geq 3$ . In lines 7 ~ 13, 16 ~ 17, 25 ~ 30, and 32 ~ 33 of Algorithm N2SMain, we have

$$T(n) \leq T(n - 1) + O(n). \quad (3.2)$$

In lines 45 ~ 47 of Algorithm N2SMain, we have

$$T(n) \leq T(n - 1) + O(n^2). \quad (3.3)$$

In lines 19 ~ 20, 35 ~ 42, and 49 ~ 52 of Algorithm N2SMain, we have

$$T(n) \leq 2T(n - 1) + O(n^2). \quad (3.4)$$

Thus, based on Eqs. (3.1)~(3.4) and Definition 1, we have

$$\begin{aligned} T(n) &\leq \max\{2T(n - 1) + O(n^2), T(n - 1) + O(n^2), T(n - 1) + O(n)\} \\ &\leq \max\{O(n^2 2^n), O(n^3), O(n^2)\} \\ &\leq O(n^2 2^n) \\ &= O(N \log^2 N). \end{aligned} \quad (3.5)$$

where  $N = 2^n$ . Therefore, according to Eq. (3.5), under the worst case, the time complexity of algorithm N2SMain is  $T(n) \leq O(N \log^2 N)$ , where  $n \leq 2$  and  $N = 2^n$  is the node number of  $C_n$ .

### 3.3 Evaluation of algorithm $N2SM_{\text{Main}}$

In this subsection, we use simulations to evaluate the performance of Algorithm  $N2SM_{\text{Main}}$ . The Algorithm  $N2SM_{\text{Main}}$  was implemented by using the programming language Python under version 3.8.1 with libraries include Networkx, Matplotlib, NumPy, Cython, etc. The program was run on a Lenovo notebook computer with Intel 1.61GHz CPU, 16GB DRAM, and 512GB hard disk.

In our simulations, A set of  $n + 1$  destination nodes  $D = \{d_1, d_2, \dots, d_{n+1}\}$  are selected randomly. Then, a source node  $s$  randomly was selected with  $s \notin D$ . Furthermore, we applied Algorithm  $N2SM_{\text{Main}}$  and measured the execution time and the path length. The results were obtained by at least 10000 simulation runs.

Given a  $n$ -dimensional cross-cube,  $C_n$ , we use  $M(n)$  and  $D(n)$  to denote the maximal length of disjoint paths constructed by Algorithm  $N2SM_{\text{Main}}$  and diameter of  $C_n$ , respectively. In the following, a comparison of  $M(n)$  and  $D(n)$  is shown in Fig. 12. In the experiment, we can find that the maximal length of node-to-set disjoint paths gotten by our algorithm is much closer to the diameter of  $C_n$ .

## 4 Conclusions

Cross-cubes have been proposed as significant variations of hypercubes, which have higher fault-tolerant capability and smaller diameter than hypercubes at the same dimension. In this paper, we construct node-to-set disjoint paths of an  $n$ -dimensional cross-cube,  $C_n$ , whose maximum length is limited by  $2n - 3$ . Furthermore, we propose an  $O(N \log^2 N)$  algorithm for finding node-to-set disjoint paths of  $C_n$ , where  $N$  is the node number of  $C_n$ . Then, we give the simulation results of the maximal length of disjoint paths obtained by our algorithm. Last but not least, as approaching to the end of this paper, further research issues on the cross-cubes are suggested. Some of them are fairly intriguing and still open for cross-cubes, e.g., set-to-set disjoint paths [21, 22], disjoint path covers [23, 24], completely independent spanning trees [25], and fault-tolerant routing [26].

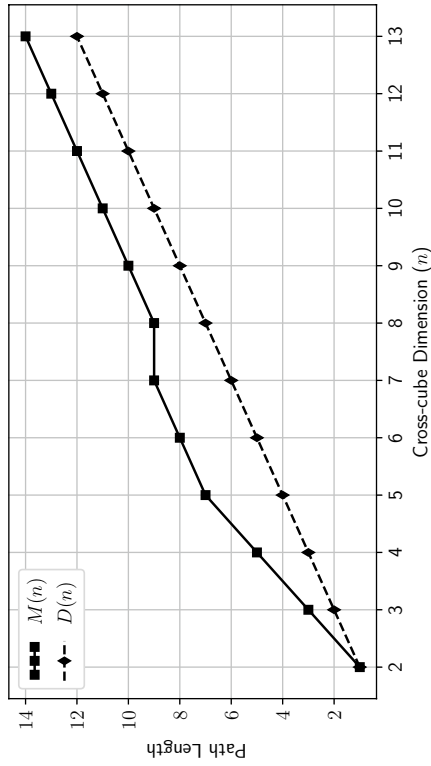


Fig. 12 A comparison of  $M(n)$  and  $D(n)$  in  $C_n$

**Acknowledgements** This paper was supported by the National Natural Science Foundation of China (Nos. 61702351 and 61572337), the Joint Found of the National Natural Science Foundation of China (No. U1905211), the Natural Science Foundation of Jiangsu Province (No. BK20180209), the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (No. 18KJD520004), the Jiangsu Planned Projects for Postdoctoral Research Funds (No 1701172B), the Fundamental Research Funds for the Central Universities of Jilin University (No 93K172020K25), and the Qing Lan Project of Jiangsu Province.

## References

1. Fu H, Liao J, Yang J et al (2016) The sunway TaihuLight supercomputer: system and applications. *Sci China Inform Sci* 59(7):072001
2. Harary F, Hayes JP, Wu H-J (1988) A survey of the theory of hypercube graphs. *Comput Math Appl* 15(4):277–289
3. Haq E (1991) Cross-cube: a new fault tolerant hypercube-based network. In: *Proceedings of fifth international parallel processing symposium*, pp 471–474
4. Efe K (1992) The crossed cube architecture for parallel computation. *IEEE Comput Archit Lett* 3(05):513–524
5. Cull P, Larson SM (1995) The Möbius cubes. *IEEE Trans Comput* 44(5):647–659
6. Zhou W, Fan J, Jia X, Zhang S (2011) The spined cube: a new hypercube variant with smaller diameter. *Inform Process Lett* 111(12):561–567
7. Wang X, Fan J, Jia X, Lin C-K (2016) An efficient algorithm to construct disjoint path covers of DCell networks. *Theor Comput Sci* 609(1):197–210
8. Wang X, Fan J, Lin C-K, Jia X (2016) Vertex-disjoint paths in DCell networks. *J Parall Distrib Comput* 96:38–44
9. Kaneko K, Bossard A (2017) A set-to-set disjoint paths routing algorithm in tori. *Int J Netw Comput* 7(2):173–186
10. Bossard A, Kaneko K (2014) Time optimal node-to-set disjoint paths routing in hypercubes. *J Inform Sci Eng* 30(4):1087–1093
11. Kocik D, Hirai Y, Kaneko K (2016) Node-to-set disjoint paths problem in a Möbius cube. *IEICE Trans Inform Syst* 99(3):708–713
12. Gu Q, Peng S (1997) Node-to-set disjoint paths problem in star graphs. *Inform Process Lett* 62(4):201–207
13. Kaneko K, Suzuki Y (2003) Node-to-set disjoint paths problem in pancake graphs. *IEICE Trans Inform Syst* 86(9):1628–1633
14. Kaneko K (2003) An algorithm for node-to-set disjoint paths problem in burnt pancake graphs. *IEICE Trans Inform Syst* 86(12):2588–2594
15. Bossard A, Kaneko K (2012) Node-to-set disjoint-path routing in hierarchical cubic networks. *Comput J* 55(12):1440–1446
16. Bossard A, Kaneko K, Peng S (2011) A new node-to-set disjoint-path algorithm in perfect hierarchical hypercubes. *Comput J* 54(8):1372–1381
17. Bossard A, Kaneko K, Peng S (2011) Node-to-set disjoint-path routing in perfect hierarchical hypercubes. *Procedia Comput Sci* 4:442–451
18. Ling S, Chen W (2013) Node-to-set disjoint paths in biswapped networks. *Comput J* 57(7):953–967
19. Diestel R (2012) *Graph theory*, 4th edn, vol 173. In: *Graduate texts in mathematics*. Springer
20. Wang X, He F, Zhang S (2018) A restricted fault-free unicast algorithm in cross-cubes. *J Southwest China Norm Univ* 43(9):57–65 in Chinese
21. Bossard A, Kaneko K (2012) The set-to-set disjoint-path problem in perfect hierarchical hypercubes. *Comput J* 55(6):769–775
22. Bossard A, Kaneko K (2016) Set-to-set disjoint paths routing in torus-connected cycles. *IEICE Trans Inform Syst* 99(11):2821–2823
23. Lü H (2019) Paired many-to-many two-disjoint path cover of balanced hypercubes with faulty edges. *J Supercomput* 75(1):400–424



24. Park J-H, Kim J-H, Lim H-S (2019) Disjoint path covers joining prescribed source and sink sets in interval graphs. *Theor Comput Sci* 776:125–137
25. Cheng B, Wang D, Fan J (2017) Constructing completely independent spanning trees in crossed cubes. *Disc Appl Math* 219:100–109
26. Fan J, Jia X, Cheng B, Yu J (2011) An efficient fault-tolerant routing algorithm in bijective connection networks with restricted faulty edges. *Theor Comput Sci* 412(29):3440–3450

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.