d-Separated Paths in Hypercubes and Star Graphs

Qian-Ping Gu and Shietung Peng
Department of Computer Software, The University of Aizu
Aizu-Wakamatsu, Fukushima, 965-80 Japan
e-mail: {qian/s-peng}@u-aizu.ac.jp

Abstarct: *In this paper, we consider a generalized node disjoint paths problem: d-separated paths problem. In a graph G, given two distinct nodes s and t, two paths P and Q, connecting s and t, are d-separated if* $d_{G-\{s,t\}}(u,v) \geq d$ *for any* $u \in P - \{s,t\}$ *and* $v \in Q - \{s,t\}$*, where* $d_{G-\{s,t\}}(u,v)$ *is the distance between u and v in the reduced graph* $G - \{s,t\}$*. d-separated paths problem is to find as many d-separated paths between s and t as possible. In this paper, we give the following results on d-separated paths problems on n-dimensional hypercubes* H_n *and star graphs* G_n*. Given s and t in* H_n*, there are at least* $(n-2)$ *2-separated paths between s and t.* $(n-2)$ *is the maximum number of 2-separated paths between s and t for* $d(s,t) \geq 4$*. Moreover,* $(n-2)$ *2-separated paths of length at most* $\min\{d(s,t)+2, n+1\}$ *for* $d(s,t) < n$ *and of length n for* $d(s,t) = n$ *between s and t can be constructed in* $O(n^2)$ *optimal time. For* $d \geq 3$*, d-separated paths in* H_n *do not exist. Given s and t in* G_n*, there are exactly* $(n-1)$ *d-separated paths between s and t for* $1 \leq d \leq 3$*.* $(n-1)$ *3-separated paths of length at most* $\min\{d(s,t)+4, d(G_n)+2\}$ *between s and t can be constructed in* $O(n^2)$ *optimal time, where* $d(G_n) = \lfloor \frac{3(n-1)}{2} \rfloor$*. For* $d \geq 5$*, d-separated paths in* G_n *do not exist.*

Key words: Interconnection networks, node disjoint paths, graph algorithms,

1. Introduction

Finding disjoint paths (refer node disjoint paths in this paper) in interconnection networks is one of the fundamental issues in design and implementation of parallel and distributed computing systems [9, 8, 2, 15, 16, 18, 14, 4, 6]. Finding multiple disjoint paths is also one of major approaches to solve fault tolerant routing problem. For example, given two nodes s and t in a graph, k disjoint paths between s and t guarantee the routing path $s \rightarrow t$ exists in the presence of up to $k-1$ arbitrary faulty nodes. For a specific routing problem, we say a graph can tolerate l faulty nodes if given at most l arbitrary faulty nodes, the required routing path exists for the routing problem. In the case of more than l faulty nodes, if the routing path does not always exist, what are the conditions for the existence of the routing path? This problem has been attracting much attention and several approaches such as *forbidden faulty set*, *cluster fault tolerant routing*, and so on have been developed [12, 10, 5, 7].

Recently, fault tolerant routing was studied for the case that faulty nodes can be covered by subgraphs of small diameters [5, 7]. It has been shown that several important interconnection networks such as hypercubes, star graphs, and so on, can tolerate l arbitrary faulty clusters of small diameters rather than l arbitrary faulty nodes, where a faulty cluster is a connected subgraph such that all its nodes are faulty [5, 7]. In practice, processors of an interconnection network often fail in a *cluster-like* manner. For example, two routing jobs, one is between nodes s_1 and t_1 and the other is between s_2 and t_2, are performed simultaneously in a graph. The nodes in the routing path between s_1 and t_1 can be viewed as faulty nodes by the routing between s_2 and t_2, and vice versa.

To develop a new approach for solving fault tolerant routing, in this paper, we study a generalized disjoint paths problem, d-separated paths problem: In a graph G, given two distinct nodes s and t, two paths P and Q, connecting s and t, are d-separated if $d_{G-\{s,t\}}(u,v) \geq d$ for any $u \in P - \{s,t\}$ and $v \in Q - \{s,t\}$, where $d_{G-\{s,t\}}(u,v)$ is the distance between u and v in the reduced graph $G - \{s,t\}$. d-separated paths problem is to find as many d-separated paths between s and t as possible. When $d = 1$, d-separated paths problem becomes conventional node-to-node disjoint paths problem. For two nodes s and t, k d-separated paths between s and t guarantee the routing path $s \rightarrow t$ exists if the faulty nodes can be covered by at most $k-1$ subgraphs of diameter at most $d-1$, even though the total number of faulty nodes are

[0]This research was partially supported by The Founding of Group Research Projects at The University of Aizu

far beyond $k-1$. Therefore, d-separated paths with a larger d would make the communication between s and t more reliable.

Hypercubes and star graphs are interesting interconnection networks for parallel computation and communication. A number of efficient algorithms for disjoint paths problems and fault tolerant routings in hypercubes and star graphs have been proposed [18, 3, 13, 11, 17, 5, 7]. It is previously known that for s and t in n-dimensional hypercubes H_n, there are n disjoint paths between s and t of length at most $\min\{d(s,t)+2, n+1\}$ for $d(s,t) < n$ and of length n for $d(s,t) = n$ [18]. For s and t in n-dimensional star graphs G_n, there are $n-1$ disjoint paths of length at most $\min\{d(s,t)+4, d(G_n)+1\}$ between s and t [13, 3, 11, 17], where $d(G_n) = \lfloor\frac{3(n-1)}{2}\rfloor$ is the diameter of G_n. The bounds on the lengths of disjoint paths given in [18, 13, 3, 11, 17] are optimal. In this paper, we investigate d-separated paths problem in H_n and G_n. Given distinct nodes s and t in H_n, we prove that there are at least $(n-2)$ 2-separated paths between s and t. We also show that $(n-2)$ is the maximum number of 2-separated paths between s and t for $d(s,t) \geq 4$. We give an algorithm which, given s and t in H_n, constructs $(n-2)$ 2-separated paths between s and t of length at most $\min\{d(s,t)+2, n+1\}$ for $d(s,t) < n$ and of length n for $d(s,t) = n$ in $O(n^2)$ optimal time. In fact our algorithm can construct n disjoint paths between s and t with a subset of $(n-2)$ paths are 2-separated. Since the distance between two arbitrary neighbors of s in the reduced graph $H_n - \{s,t\}$ is 2, for $d \geq 3$, d-separated paths in H_n do not exist. Given s and t in G_n, we show that there are $(n-1)$ 3-separated paths between s and t. Since G_n is $(n-1)$-connected, $(n-1)$ is the maximum number of d-separated paths between s and t for $1 \leq d \leq 3$. We also give an algorithm for constructing $(n-1)$ 3-separated paths of length at most $\min\{d(s,t)+4, d(G_n)+2\}$ between s and t in $O(n^2)$ optimal time. For any two neighbors u and v of a node s in G_n, $d_{G_n-\{s\}}(u,v) = 4$. Thus, for $d \geq 5$, d-separated paths in G_n do not exist. The d-separated paths given in this paper for hypercubes and star graphs enhance the fault tolerant properties of hypercubes and star graphs.

The rest of this paper is organized into three sections. d-separated paths in hypercubes and star graphs are discussed in Sections 2 and 3, respectively. Some concluding remarks and open problems are given in the last section.

2. d-Separated Paths in Hypercubes

A path in a graph G is a sequence of edges of the form $(s_1,s_2)(s_2,s_3)\ldots(s_{k-1},s_k)$, $s_i \in G$, $1 \leq i \leq k$, and $s_i \neq s_j$, $i \neq j$. The length of a path is the number of edges in the path. We sometimes denote the path from s_1 to s_k by $s_1 \to s_k$. For a path $P = (s_1,s_2)\ldots(s_{k-1},s_k)$, we also use P to denote the set $\{s_1,\ldots,s_k\}$ of nodes that appear in path P if there is no confusion arises. Given two nodes $s,t \in G$, $d(s,t)$ denotes the distance between s and t, i.e., the length of the shortest path connecting s and t, in graph G, and $d_{G-\{s,t\}}(s,t)$ denotes the distance between s and t in the reduced graph $G-\{s,t\}$. The diameter of G is defined as $d(G) = \max\{d(s,t) | s,t \in G\}$. In this paper, $<n> = \{1,2,\ldots,n\}$. For sets A and B, we will use $A-B$ to denote the set $C = \{x | x \in A \text{ and } x \notin B\}$. Given two nodes s and t in a graph G, paths P and Q between s and t are called *d-separated* if for any nodes $u \in P-\{s,t\}$ and $v \in Q-\{s,t\}$, $d_{G-\{s,t\}}(u,v) \geq d$. We call a set of paths d-separated if any two paths of the set are d-separated. For $d=1$, d-separated paths are called *disjoint paths.*

An n-dimensional hypercube H_n is an undirected graph on node set $\{0,1\}^n$ such that there is an edge between $u \in H_n$ and $v \in H_n$ iff u and v differ exactly in one bit position. H_n is node and edge symmetric, is n-connected, and has diameter $d(H_n) = n$.

We will adopt the following notations in the discussion of this section. For a node $s = a_1a_2\ldots a_n \in H_n$, $s^{(i)}$, $1 \leq i \leq n$, denotes the node $a_1\ldots a_{i-1}\bar{a}_i a_{i+1}\ldots a_n$, where $\bar{a}_i$ is the logical negation of a_i. Similarly, $s^{(i_1,i_2,\ldots,i_k)}$ denotes the node $b_1\ldots b_n$, where $b_{i_j} = \bar{a}_{i_j}$, $1 \leq j \leq k$, and $b_l = a_l$ for $l \in <n> - \{i_1,\ldots,i_k\}$. Given two nodes s and t in H_n and k disjoint paths $P_1,\ldots,P_k$ between s and t, these k paths pass through k neighbors $s^{(i_1)},\ldots,s^{(i_k)}$ of s and k neighbors $t^{(j_1)},\ldots,t^{(j_k)}$ of t. We call the set $S = \{i_1,\ldots,i_k\}$ ($T = \{j_1,\ldots,j_k\}$) the *port-set* of s (t) through which paths $P_1,\ldots,P_k$ pass.

For any two neighbors u and v of a node s in H_n, obviously $d_{H_n-\{s,t\}}(u,v) = 2$. Thus, d-separated paths for $d \geq 3$ do not exist in H_n. For $d=1$, d-separated paths in H_n were enumerated in [18]. Now, we find 2-separated paths in H_n.

Theorem 1 *Given two nodes s and t in H_n with $d(s,t) = n \geq 4$, $(n-2)$ 2-separated paths of length n between s and t can be found in $O(n^2)$ time.*

Proof: To prove the theorem, we show the following statement: given s and t in H_n with $d(s,t) = n$, where $n \geq 4$ and $n \neq 5$, and any two integers p and q of $<n>$, l ($l \leq n-2$) 2-separated paths of length n between s and t that

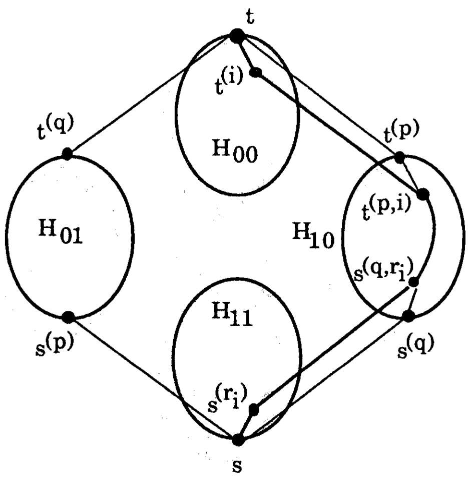

Figure 1: Partition H_n into H_{00}, H_{01}, H_{10}, and H_{11}.

pass through port-sets $S = T \subset (< n > -\{p,q\})$ ($|S| = l$) can be found. Since H_n is node symmetric, we can assume $t = 00\ldots0$ and $s = 11\ldots1$. The statement is proved by induction on n.

To show the induction base, without loss of generality, we assume $p = n-1$ and $q = n$. For $n = 4$, we can find two 2-separated paths as:

$$P_1 \; : \; t \to 1000 \to 1001 \to 1011 \to s$$
$$P_2 \; : \; t \to 0100 \to 0110 \to 0111 \to s.$$

Let $n = 6$. The argument for $n = 4$ above implies the case of $l = 2$. For $l = 3$ and $S = \{1,2,3\}$, we can find the three 2-separated paths as:

$$\begin{aligned} P_1 : t &\to 100000 \to 100100 \to 110100 \\ &\to 110101 \to 110111 \to s \\ P_2 : t &\to 010000 \to 010010 \to 011010 \\ &\to 011011 \to 011111 \to s \\ P_3 : t &\to 001000 \to 001001 \to 101001 \\ &\to 101011 \to 101111 \to s. \end{aligned}$$

For $l = 4$, partition H_n into 4 $(n-2)$-dimensional subcubes H_{00}, H_{01}, H_{10}, and H_{11}, such that the two bits p and q of the four subcubes are 00, 01, 10, and 11, respectively (see Figure 1). Obviously $t \in H_{00}$, $t^{(p)} \in H_{10}$, $t^{(q)} \in H_{01}$, $s \in H_{11}$, $s^{(p)} \in H_{01}$, and $s^{(q)} \in H_{10}$. We find two 2-separated paths which pass through port-sets $S_{10} = T_{10} = \{1,2\}$ and H_{10}, and find two 2-separated paths which pass through port-sets $S_{01} = T_{01} = \{3,4\}$ and H_{01}. Obviously, the above four paths are 2-separated.

Let $n = 7$, the argument for $n = 6$ implies the statement if $l \leq 4$. For $l = 5$, we can construct three 2-separated paths which pass through H_{10} and two 2-separated paths which pass through H_{01} (in the following paths $P = 10$ and $Q = 01$) as:

$$\begin{aligned} P_1 : t &\to 1000000 \to 10000P \to 10001P \\ &\to 10011P \to 11011P \to 1101111 \to s \\ P_2 : t &\to 0100000 \to 01000P \to 01001P \\ &\to 01101P \to 11101P \to 1110111 \to s \\ P_3 : t &\to 0010000 \to 00100P \to 00110P \\ &\to 10110P \to 11110P \to 1111011 \to s \\ P_4 : t &\to 0000100 \to 00001Q \to 01001Q \\ &\to 01011Q \to 01111Q \to 0111111 \to s \\ P_2 : t &\to 0001000 \to 00010Q \to 10010Q \\ &\to 10011Q \to 10111Q \to 1011111 \to s. \end{aligned}$$

Assume the statement holds for $k \leq n-1$, $k \geq 7$, and we prove it for n. Let p and q be any two integers of $< n >$. Partition H_n into 4 $(n-2)$-dimensional subcubes H_{00}, H_{01}, H_{10}, and H_{11} as we did above. Let $l = \lfloor (n-2)/2 \rfloor$, $S_{10} = T_{10}$ be a subset of $< n > -\{p,q\}$ with $|S_{10}| = l$, and $S_{01} = T_{01} = (< n > -\{p,q\}) - S_{10}$. From the induction hypothesis, we can construct l 2-separated paths between $t^{(p)}$ and $s^{(q)}$ of length $n-2$ that pass through port-sets S_{10} and T_{10} in H_{10}, and construct $m = (n-2-l)$ 2-separated paths between $t^{(q)}$ and $s^{(p)}$ of length $n-2$ that pass through port-sets S_{01} and T_{01} in H_{01}. After this, for each path $t^{(p)} \to t^{(p,i)} \to s^{(r_i,q)} \to s^{(q)}$, disconnect $t^{(p,i)}$ from $t^{(p)}$ and connect $t^{(p,i)}$ to $t^{(i)}$, and disconnect $s^{(r_i,q)}$ from $s^{(q)}$ and connect $s^{(r_i,q)}$ to $s^{(r_i)}$ (Figure 1). We get l 2-separated paths $P_1, \ldots, P_l$ of length n between s and t. Similarly, we can get m 2-separated paths $Q_1, \ldots, Q_m$, $m = (n-2) - l$ of length n, by reconstructing the paths from $t^{(q)}$ to $s^{(p)}$. For any path

$$P_i : t \to t^{(i)} \to t^{(i,p)} \to s^{(q,r_i)} \to s^{(r_i)} \to s,$$

$i \in T_{10}$, and any path

$$Q_j : t \to t^{(j)} \to t^{(j,q)} \to s^{(p,r_j)} \to s^{(r_j)} \to s,$$

$j \in T_{01}$, we show that P_i and Q_j are 2-separated. Since $i, r_i \in T_{10} = S_{10}$ and $j, r_j \in T_{01} = S_{01}$, $t^{(i)} \neq t^{(j)}$ and $s^{(r_i)} \neq s^{(r_j)}$. From this, $d(t^{(i)}, t^{(j)}) = 2$ and $d(s^{(r_i)}, s^{(r_j)}) = 2$. Let u be any node in the subpath $t^{(i,p)} \to s^{(q,r_i)}$ and v be any node in $t^{(j,q)} \to t^{(p,r_j)}$. Since there is no edge between subcubes H_{01} and H_{10}, $d(u,v) \geq 2$. Similarly, we can show that for any $u \in P_i - \{s,t\}$

and $v \in Q_j - \{s,t\}$, $d_{H_n-\{s,t\}}(u,v) \geq 2$. Thus, the statement holds.

To show the theorem, we still need to prove it for $n = 5$. In fact, this has been done in the proof of the statement for $n = 7$ and $l = 5$. It is easy to see that the above proof implies a recursive algorithm which constructs the $(n-2)$ 2-separated paths. It takes $O(n)$ time in each recursive step and $O(n^2)$ time in total to find the paths. □

In addition to the $(n-2)$ 2-separated paths given in Theorem 1, two more disjoint paths between s and t can be found. These paths will be used in constructing 2-separated paths for the case of $d(s,t) < n$.

Lemma 2 *For two distinct nodes s and t in H_n with $d(s,t) = n \geq 4$, n disjoint paths, among them $(n-2)$ paths are 2-separated, of length n can be constructed in $O(n^2)$ time.*

Proof Sketch: For $n \leq 5$, the lemma can be proved by an enumerate argument. For $n \geq 6$, let $S = T = < n > -\{p,q\}$. From Theorem 1, we can find $(n-2)$ 2-separated paths which pass through port-sets S and T. Partition H_n into 4 $(n-2)$-dimensional subcubes H_{00}, H_{01}, H_{10}, and H_{11} as in Theorem 1. The two additional disjoint paths can be constructed as follows. One path P starts from t to $t^{(p)}$ which is in H_{10} and the other path Q starts from t to $t^{(q)}$ which is in H_{01}. Since H_{10} has $l \leq (n-2)-2$ 2-separated paths constructed by Theorem 1, by an inductive argument, we can find two disjoint paths between $t^{(p)}$ and $s^{(q)}$ in H_{10} that are disjoint with the l 2-separated paths. Choose one of the two paths, we get path P. Similarly, path Q can be obtained. □

Theorem 3 *For two arbitrary nodes s and t in H_n with $d(s,t) = k$, $(n-2)$ 2-separated paths, among them $k-2$ are of length $d(s,t)$ and $n-k$ are of length $d(s,t)+2$, can be found in $O(n^2)$ time.*

Proof: Without loss of generality, we assume that $t = 00\ldots0$ and the first k bits and the last $n-k$ bits of s are 1 and 0, respectively. Let H_k be the k-dimensional subcube obtained by fixing the last $n-k$ bits of H_n into 0. Then $s,t \in H_k$ (see Figure 2). From Theorem 1, we can find $(k-2)$ 2-separated paths between s and t of length k in H_k. From Lemma 2, we can find a path P of length k that is disjoint with the $(k-2)$ 2-separated path. Let $H_k^{(i)}$, $k+1 \leq i \leq n$, be the k-dimensional subcube obtained by fixing the last $n-k$ bits except the ith bit to 0 and fixing the ith bit to 1. Then $t^{(i)}, s^{(i)} \in H_k^{(i)}$ and $d(s^{(i)}, t^{(i)}) = k$ (Figure 2). We map path P in H_k into $H_k^{(i)}$ as

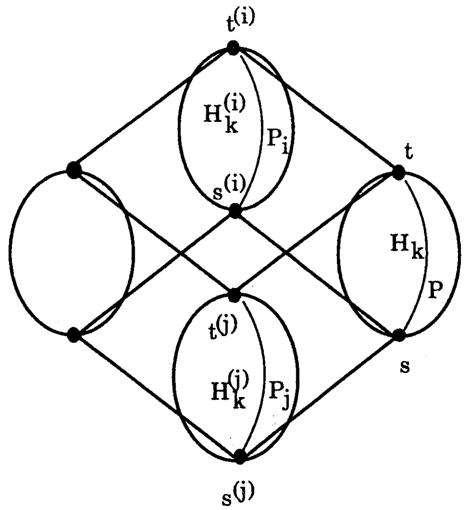

Figure 2: Partition H_n into H_k and $H_k^{(i)}$.

follows: for each node u in P, we change the ith bit of u from 0 to 1. Then, we get a path P_i of length k between $s^{(i)}$ and $t^{(i)}$ in $H_k^{(i)}$. Since P is disjoint with the $(k-2)$ 2-separated paths in H_k, P_i is 2-separated with these $k-2$ paths. Obviously, paths P_i and P_j, $k+1 \leq i,j \leq n$ and $i \neq j$, are 2-separated. The length of the path P_i is $k+2$. □

We have proved that for any $s,t \in H_n$, n disjoint paths, among them $n-2$ are 2-separated can be found. Now, we show that $(n-2)$ is the maximum number of 2-separated paths between any two nodes s and t of H_n with $d(s,t) \geq 4$.

Theorem 4 *For two nodes s and t in H_n with $d(s,t) \geq 4$, let $\mathbf{P}$ be any set of 2-separated paths between s and t. Then $|\mathbf{P}| \leq n-2$.*

Proof: We first show the case of $d(s,t) = n$. Assume there is a set $\mathbf{P} = \{P_1, \ldots, P_m\}$, $m \geq n-1$, of 2-separated paths between s and t. Furthermore, we assume path P_i passes through neighbor $t^{(i)}$ of t, $1 \leq i \leq m$. The next node connected to $t^{(i)}$ in P_i can not be $t^{(i,j)}$ for any $1 \leq j \leq m$, since $d(t^{(i,j)}, t^{(j)}) = 1$ and P_i and P_j are 2-separated. Thus, $m \leq n-1$. Assume $m = n-1$. Then $t^{(i)}$, $1 \leq i \leq n-1$, must be connected to $t^{(i,n)}$. The next node connected to $t^{(i,n)}$ must be $t^{(i,n,j)}$ for some j with $j \neq i,n$ (notice that $d(s,t) \geq 4$). However, $d(t^{(i,n,j)}, t^{(j,n)}) = 1$, a contradiction to P_i and P_j are 2-separated. Thus, $|\mathbf{P}| \leq n-2$.

Now prove the general case of $d(s,t) = k$. Let

H_k be the k-dimensional subcube which contains s and t. Then, from the above argument, we can have at most $(k-2)$ 2-separated paths between s and t in H_k. Since s has $n-k$ neighbors which are not in H_k, we can have at most $(n-k)$ 2-separated paths outside H_k. Let $t^{(p)}$ and $t^{(q)}$ be the two neighbors that have not been used by the $(n-2)$ 2-separated paths. Then obviously, there is a node u in one of the $(n-2)$ 2-separated paths such that $d(u, t^{(p,i)}) = 1$ for any $i \in < n >$. □

3. d-Separated Paths in Star Graphs

An n-dimensional star graph is a graph G_n, where the nodes of G_n are in a 1-1 correspondence with the permutations $[p_1, p_2 \ldots, p_n]$ of the set $< n >= \{1, 2, \ldots, n\}$. Two nodes of G_n are connected by an edge if and only if the permutation of one node can be obtained from the other by interchanging the first symbol p_1 with the ith symbol p_i, $2 \leq i \leq n$. This interchange of the symbols in position 1 with position i is called a *transposition*. For node $s = [p_1, p_2, \cdots, p_n]$, $s^{(i)}$ denotes the node $[p_i, p_2, \cdots, p_{i-1}, p_1, p_{i+1}, \cdots, p_n]$, obtained by transposition i on s. G_n has $n!$ nodes, and $n! \times \frac{(n-1)}{2}$ edges. It has uniform node degree $n-1$ and diameter $d(G_n) = \lfloor \frac{3(n-1)}{2} \rfloor$. G_n is node and edge symmetric and is $(n-1)$-connected.

Since star graphs are node symmetric, in node-to-node routing, node t can be assumed to have the identity permutation $I = [1, 2, \ldots, n]$. Since the label of any node in G_n is a permutation of $< n >$, $s = [p_1, p_2, \ldots, p_n]$ can be viewed as a set of cyclically ordered sets of digits $(\phi_1 \phi_2 \ldots \phi_l)$ such that the position of ϕ_j in I is occupied by ϕ_{j+1} $(p_{\phi_j} = \phi_{j+1})$ for $1 \leq j \leq l-1$ and the position of ϕ_l in I is occupied by ϕ_1 $(p_{\phi_l} = \phi_1)$. For example, $s = [4, 2, 5, 1, 3]$ can be expressed by cyclically ordered sets (41)(2)(53). We will call a cycle set a *cycle* and the cyclically order sets *cycle form*. The maximum number of cycles in a permutation of n elements is n and the minimum number is 1. When a cycle (i) has only one digit i it is the case $i = p_i$. In this paper, $\Pi = \{\pi_0, \ldots, \pi_k\}$ is used to express the cycle form, and $\pi_j = (\phi_1^j \phi_2^j \ldots \phi_{l_j}^j)$ is a cycle. The number of digits in a cycle π_j is called the length of cycle, denoted as $|\pi_j|$. Given a cycle form, cycles can appear in any order in the form, and within a cycle any cyclic shift of the sequences of digits does not change the permutation presented by the cycle form. For example, (134)(52), (341)(25), and (52)(413) present the same node [3, 5, 4, 1, 2]. For cycle $\pi_j = (\phi_1^j \phi_2^j \ldots \phi_{|\pi_j|}^j)$, $\pi_j^{(i)}$ denotes the cycle $(\phi_i^j \phi_{i+1}^j \ldots \phi_{|\pi_j|}^j \phi_1^j \ldots \phi_{i-1}^j)$. For convenience, we denote the cycle which contains

merge-then-reduce(s);
for $(0 \leq i < k)$ **do** swap(ϕ_1^i, ϕ_1^{i+1});
/* merge two cycles $(\pi_i \pi_{i-1} \ldots \pi_0)$ and (π_{i+1}) into one cycle $(\pi_{i+1} \pi_i \ldots \pi_0)$. */
/* Let $(\pi_k \pi_{k-1} \ldots \pi_1)$ be denoted by π_l.*/
for $(1 \leq j < |\pi_l|)$ **do** swap(ϕ_j^l, ϕ_{j+1}^l);

Figure 3: A merge-then-reduce subroutine.

1 as $\pi_0 = (\phi_1^0 \ldots \phi_{|\pi_0|}^0)$ with $\phi_{|\pi_0|}^0 = 1$. Obviously, the first (leftmost) digit $\phi_1^0 = p_1$. Given a node s, the cycle (i) of length 1 with $i \neq 1$ will be omitted from the cycle form for s. Let $\mu(s)$ be the number of cycles of length at least 2 in the cycle form of s, and let $m(s)$ be the total number of digits in the $\mu(s)$ cycles. Then, it was shown in [13] that $d(s, I) = \mu(s) + m(s) - 2$ if $|\pi_0| \geq 2$, $d(s, I) = \mu(s) + m(s)$ otherwise.

It was shown in [1] that a shortest path between $s = [p_1, p_2, \ldots, p_n]$ and I can be found by the following rules (referred as *shortest path rules*):
(1) If $p_1 = 1$, swap(p_1, p_i) for some $p_i \neq i$, and
(2) If $p_1 = i$, $2 \leq i \leq n$, swap(p_1, p_i) or swap(p_1, p_j), $p_j \neq j$ and $p_j \notin \pi_0$.
In rule (1), swap(p_1, p_i), $p_i \in \pi_j$ with $p_i = \phi_k^j$, merges the cycles $\pi_0 = (1)$ and π_j into the cycle $(\pi_j^{(k)} 1)$. In rule (2), swap(p_1, p_i) reduces the length of cycle π_0 by 1, and swap(p_1, p_j), $p_j \in \pi_i$ with $i \neq 0$ and $p_j = \phi_k^i$, merges cycles π_0 and π_i into one cycle $(\pi_i^{(k)} \pi_0)$. Following the shortest path rules, we now give a subroutine, merge-then-reduce, which finds a shortest path from $s = \pi_0 \ldots \pi_k$ to I (see Figure 3). Obviously, the length of the path constructed by Subroutine merge-then-reduce is $(\sum_{i=0}^{k} |\pi_i|) - 1 + k$. Notice that, given an $s^{(i)}$, either $d(s^{(i)}, I) = d(s, I) + 1$ or $d(s^{(i)}, I) = d(s, I) - 1$. An $s^{(i)}$ with $d(s^{(i)}, I) = d(s, I) - 1$ can be obtained only by the the shortest path rules. An important property of the path between $s = \pi_0 \ldots \pi_k$, where $\pi_0 = (\phi_1^0 \ldots \phi_{|\pi_0|-2}^0 \phi_{|\pi_0|-1}^0 1)$, and I found by subroutine merge-then-reduce is that all the nodes except I and its neighbor in the path has the cycle form $(\ldots \phi_{|\pi_0|-2}^0 \phi_{|\pi_0|-1}^0 1) \ldots$. So, if we can find d-separated paths from s to nodes $u = (\ldots xy1) \ldots$ and $v = (\ldots x'y'1) \ldots$ such that $d((xy1), (x'y'1)) \geq d$, then the paths from u to I and the path from v to I found by subroutine merge-then-reduce are d-separated. The next lemma gives the conditions which guarantee $d((xy1), (x'y'1)) \geq 3$.

Lemma 5 *Given two nodes* $s = (\phi_1 \ldots \phi_l 1)\Pi$

and $t = (\phi'_1 \ldots \phi'_{l'} 1)\Pi'$ *in* G_n*, with* $\phi_l \neq \phi_{l'}$*, then* $d(s,t) \geq 2$ *and* $d(s,t) = 2$ *if and only if* $\Pi = \Pi'$ *and* $(l = l' = 1$ *or* $l = l' = 2$ *and* $\{\phi_1, \phi_2\} = \{\phi'_1, \phi'_2\})$.

Proof: For nodes $s = (\phi_1 \ldots \phi_l 1)\Pi$ and $t = (\phi'_1 \ldots \phi'_{l'} 1)\Pi'$ with $\phi_l \neq \phi'_{l'}$, let $s = [p_1, \ldots, p_n]$ and $t = [q_1, \ldots, q_n]$. Then $p_1 \neq 1$, $q_1 \neq 1$, and 1 appears in different positions of $[p_1, \ldots, p_n]$ and $[q_1, \ldots, q_n]$. Therefore, at least two swaps are needed to move 1 from the position in $[p_1, \ldots, p_n]$ to the position in $[q_1, \ldots, q_n]$, i.e., $d(s,t) \geq 2$. For $s = (\phi_1 1)\Pi$ and $t = (\phi_2 1)\Pi$, $s \rightarrow (1)\Pi \rightarrow (\phi_2 1)\Pi = t$ is a path between s and t. For $s = (\phi_1 \phi_2 1)\Pi$ and $t = (\phi_2 \phi_1 1)\Pi$, $s \rightarrow (1)(\phi_1\phi_2)\Pi \rightarrow (\phi_2\phi_1 1)\Pi = t$ is a path between s and t. Therefore, for $s = (\phi_1 \ldots \phi_l 1)\Pi$ and $t = (\phi'_1 \ldots \phi'_{l'} 1)\Pi'$ with $\phi_l \neq \phi'_{l'}$, $d(s,t) = 2$ if $\Pi = \Pi'$ and $(l = l' = 1$ or $l = l' = 2$ and $\{\phi_1, \phi_2\} = \{\phi'_1, \phi'_2\})$. Now, we prove $d(s,t) \geq 3$ if the condition: $\Pi = \Pi'$ or $(l = l' = 1$ or $l = l' = 2$ and $\{\phi_1, \phi_2\} = \{\phi'_1, \phi'_2\})$, does not hold. Let $t = (\phi'_1 \ldots \phi'_{l'} 1)\Pi'$ with $\Pi \neq \Pi'$ and $u = (\phi'_1 \ldots \phi'_{l'} 1)\Pi$. Then, obviously, $d(s,t) > d(s,u) \geq 2$. Assume $s = (\phi_1 1)$ and $t = (\phi_i \phi_2 1)$, $\phi_i \neq \phi_2$. Then obviously $d(s,t) = 3$. Assume $s = (\phi_i \phi_1 1)$ and $t = (\phi_j \phi_2 1)$, $\phi_i \neq \phi_1$, $\phi_j \neq \phi_2$, and $\{\phi_i, \phi_1\} \neq \{\phi_j, \phi_2\}$. Then $d(s,t) = 3$ if $\phi_i = \phi_2$ or $\phi_j = \phi_1$ and $d(s,t) = 4$ otherwise. Similarly, we can show $d(s,t) \geq 3$ if $l \geq 3$ and $l' \geq 2$ or $l \geq 2$ or $l' \geq 3$. □

Now, we give two algorithms which find $(n-1)$ 3-separated paths of length at most $d(G_n) + 2$ between $s = [p_1, p_2, \ldots, p_n]$ and I in G_n. The key point in the algorithms is to find $n-1$ 3-separated paths $s \rightarrow u_i$, where $u_i = (\ldots x_i y_i 1)\Pi_i$ and for $i \neq j$, $y_i \neq y_j$ and $\{x_i, y_i\} \neq \{x_j, y_j\}$. (**)
Once u_i is found, and then $(n-1)$ 3-separated paths can be found by applying merge-then-reduce subroutine on u_i. The first algorithm, given in Figure 4, deals with the case of $|\pi_0| = 1$ ($\pi_0 = (1)$) and the second algorithm, given in Figure 5, solves the case of $|\pi_0| \geq 2$.

Lemma 6 *Any non-trivial path from a node* s *to itself in* G_n *has the even length at least 6.*

Proof: Obviously, any non-trivial path from s to itself in G_n has an even length. For node $s \in G_n$, $s \rightarrow s^{(i)} \rightarrow s^{(i,j)} \rightarrow s^{(i,j,i)} \rightarrow s^{(i,j,i,j)} \rightarrow s^{(i,j,i,j,i)} \rightarrow s^{(i,j,i,j,i,j)} = s$, $2 \leq i \neq j \leq n$, is a non-trivial path of the minimum length. □

Lemma 7 *Given* $s = \pi_0\pi_1 \ldots \pi_k$ *with* $|\pi_i| \geq 2$, $0 \leq i \leq k$*, and there exists a* $j \neq 1$ *with* $p_j = j$ *in the permutation* $[p_1, \ldots, p_n]$ *of* s*, then* $d(s,I) \leq d(G_n) - 2$*, and* $d(s,I) = d(G_n) - 2$ *if and only if* $k + 1 = \lfloor \frac{n-1}{2} \rfloor$.

Proof: First, assume $n = 2l$. The longest distance from s to I occurs when s contains $l-2$ cycles of length 2 and one cycle of length 3. Since $d(s,I) = \mu(s) + m(s) - 2$ for $|\pi_0| \leq 2$, $d(s,I) = 3l - 4$. Therefore, $d(G_n) - 2 = (3l-2) - 2 = d(s,I)$. Next, assume $n = 2l + 1$. The longest distance from s to I occurs when s contains l cycles of length 2, In this case, we have $d(s,I) = 3l - 2$. Therefore, $d(G_n) - 2 = 3l - 2 = d(s,I)$. □

Theorem 8 *Given* $s = \pi_0\pi_1 \ldots \pi_k$ *with* $|\pi_0| = 1$ *in* G_n*, Algorithm I generates* $(n-1)$ *3-separated paths of length at most* $\min\{d(s,t) + 2, d(G_n)\}$ *from* s *to* I *in* $O(n^2)$ *time.*

Proof: Let

$$P : s \rightarrow (\pi_i 1)\pi_{i+1} \ldots \rightarrow u_{i_1} = (\phi_2^i 1)\pi_{i+1} \ldots \rightarrow I$$

and

$$Q : s \rightarrow u_{i_2} = (\pi_i^{(2)} 1)\pi_1 \ldots = (\phi_2^i \phi_1^i 1) \ldots \rightarrow I$$

be two paths from s to I generated by Algorithm I. From Lemma 6, for any two neighbors u and v of a node s in G_n, $d_{G_n - \{s\}}(u,v) = 4$. Therefore, to prove P and Q are 3-separated it is enough to show that, for any node $u \in P - \{s, s_i, I_i, I\}$ and $v \in Q - \{s, s_j, I_j, I\}$, where s_i, s_j are neighbors of s and I_i, I_j are neighbors of I, $d(u,v) \geq 3$. Since u contains cycle $(\ldots \phi_2^i 1)$ with ϕ_1^i does not belong this cycle, and v contains cycle $(\ldots \phi_2^i \phi_1^i 1)$, from Lemma 5, $d(u,v) \geq 3$. Therefore, P and Q are 3-separated. Let

$$Q : s \rightarrow (\pi_j 1)\pi_{j+1} \ldots \rightarrow u_{j_1} = (\phi_2^j 1)\pi_{j+1} \ldots \rightarrow I,$$

$j \neq i$. Then the node u has the cycle $(\ldots \phi_2^{i+1} \phi_2^i 1)$ and node v has the cycle $(\ldots \phi_2^{j+1} \phi_2^j 1)$. From Lemma 5, $d(u,v) \geq 3$ and P and Q are 3-separated. Similarly, we can show any two of the $n-1$ paths constructed in Algorithm I are 3-separated.

Obviously, Algorithm I finds the above $n-1$ paths in $O(n^2)$ time. The length of the paths found in Step 1 and Step 2 of Algorithm II are at most $d(s,I)$ and $d(s,I) + 2$, respectively, since only one move in Step 2 does not follow the shortest paths rules. However, Step 2 is executed only if there is a $p_j \neq j$ in the permutation of s, which, from Lemma 7, implies $d(s,I) \leq d(G_n) - 2$. Thus, the lemma holds. □

Theorem 9 *Given* $s = \pi_0\pi_1 \ldots \pi_k$ *with* $|\pi_0| \geq 2$ *in* G_n*, Algorithm II generates* $(n-1)$ *3-separated paths of length at most* $\min\{d(s,t) + 4, d(G_n) + 2\}$ *from* s *to* I *in* $O(n^2)$ *time.*

Proof: Let $P : s \to u_i = (\phi_1^0 \dots \phi_{i-1}^0 1)\pi_1 \dots \pi_k \to I$ and $Q : s \to u_j = (\phi_1^0 \dots \phi_{j-1}^0 1)\pi_1 \dots \pi_k \to I$, where $2 \le i < j \le |\pi_0|$, be any two paths constructed in Step 1 of Case 2. For any node $u \in \{u_i \to I\} - \{I_i, I\}$ and $v \in \{u_j \to I\} - \{I_j, I\}$, u contains cycle $(\dots \phi_{i-1}^0 1)$ and v contains $(\dots \phi_{i-1}^0 \dots \phi_{j-1}^0 1)$. From Lemma 5, $d(u,v) \ge 3$. For $3 \le i \le |\pi_0| - 1$, the nodes in the paths $\{s \to u_i\} - \{u_i\}$ contain cycles $(\phi_{i'}^0 \dots \phi_{|\pi_0|-1}^0 1)(\phi_1^0 \dots \phi_{i-1}^0)$, $i < i'$. Since ϕ_i^0 is removed first in the reduction of $(\phi_i^0 \dots \phi_{|\pi_0|-1}^0 1)$ to (1), from $|\pi_i| < |\pi_j|$, for $u \in \{s \to u_i\}$ and $v \in \{s \to u_j\}$ with $i < j$, $d(u,v) \ge 3$, if $u \ne s$, $v \ne s$, and u and v are not neighbors of s. Similarly, we can show that any two of the $n-1$ paths constructed in Algorithm II are 3-separated.

The length of the paths found in Step 1, Step 2, and Step 3 of Algorithm II is at most $d(s,I)$, $d(s,I)+2$, and $d(s,I)+4$, respectively. Step 3 is executed if there is a $p_j = j$. From Lemma 7, $d(s,I) \le d(G_n) - 2$, and thus, the lemma holds. □

4. Conclusional Remarks

There are many interesting open problems in this new subject, especially, d-separated paths in hypercubes and star graphs. We have shown 3-separated paths in G_n and G_n do not have d-separated paths for $d \ge 5$. How many 4-separated paths are there in G_n? Another open problem is that given k arbitrary 2-separated (3-separated) paths in H_n (G_n), how many additional 2-separated paths (3-separated paths) can be found? Investigating d-separated paths in other graphs or interconnection networks is surely worth further research attention.

References

[1] S. B. Akers, D. Harel, and B. Krishnamurthy. The star graph: an attractive alternative to the n-cube. In *Proc. Inthernational Conference on Parallel Processing*, pages 393–400, 1987.

[2] J. C. Bermond. Interconnection networks. *Discrete Math.*, Special issue, 1992.

[3] K. Day and A. Tripathi. A comparative study of topological properties of hypercubes and star graphs. *IEEE Trans. on Parallel and Distributed Systems*, 5(1):31–38, 1994.

[4] M. Dietzfelbinger, S. Madhavapeddy, and I. H. Sudborough. Three disjoint path paradigms in star networks. In *Proc. IEEE Symposium on Parallel and Distributed Processing*, pages 400–406, 1991.

[5] Q. Gu and S. Peng. k-pairwise cluster fault tolerant routing in hypercubes. In *Proc. of the 5th International Symposium on Algorithms and Computation ISAAC'94*, pages 345–353, 1994.

[6] Q. Gu and S. Peng. An efficient algorithm for k-pairwise node disjoint path problem in hypercubes. In *Proc. of the 7th IEEE Symposium on Parallel and Distributed Processing (SPDP'95)*, page To appear, 1995.

[7] Q. Gu and S. Peng. Node-to-node cluster fault tolerant routing in star graphs. *Information Processing Letters*, To appear.

[8] D. F. Hsu. Interconnection networks and algorithms. *Networks*, Special issue, 1993.

[9] D. F. Hsu. On container with width and length in graphs, groups, and networks. *IEICE Trans. on Fundamental of Electronics, Information, and Computer Sciences*, E77-A(4):668–680, 1994.

[10] S. Hu and C. Yang. Fault tolerance on star graphs. In *Proc. of the First International Symposium on Parallel Algorithms/Architectures Synthesis (pAs'95)*, pages 176–182, 1995.

[11] Z. Jovanovic and J. Misic. Fault tolerance of the star graph interconnection network. *Information Processing Letters*, 49:145–150, 1994.

[12] S. Latifi. Combinatorial analysis of the fault diameter of the n-cube. *IEEE Trans. on Computers*, 42(1):27–33, 1993.

[13] S. Latifi. On the fault-diameter of the star graph. *Information Processing Letters*, 46:143–150, 1993.

[14] S. Madhavapeddy and I. H. Sudborough. A topological property of hypercubes: Node disjoint paths. In *Proc. 2nd IEEE Symp. on Parallel and Distributed Processing*, pages 532–539, 1990.

[15] J. McHugh. *Algorithmic Graph Theory*. Prentice-Hall Inc., 1990.

[16] M. A. Rabin. Efficient dispersal of information for security, load balancing, and fault tolerance. *J. ACM*, Vol. 36, No. 2:335–348, 1989.

[17] Y. Rouskov and P.K. Srimani. Fault diameter of star graphs. *Information Processing Letters*, 48:234–251, 1993.

[18] Y. Saad and M. H. Shultz. Topological properties of hypercubes. *IEEE Trans. on Computers*, C-37, No. 7:867–872, 1988.

Algorithm I
Input: A node $s = \pi_0\pi_1 \ldots \pi_k$ with $|\pi_0| = 1$ in G_n.
Output: $(n-1)$ 3-separated paths from s to I.
begin /* Assume l cycles $\pi_1 \ldots \pi_l$, $0 \leq l \leq k$, are of length 2.
To get paths $s \rightarrow u_r = (\ldots x_r y_r 1)\Pi_r$ which satisfies (**), swap$(1, \phi^i_j)$ for every π_i are needed.
However, if there are at least two π_i with $|\pi_i| = 2$ $(l \geq 2)$,
special cares are needed before merge-then-reduce is executed. */
1. **if** $(l \leq 1)$ **then for** $(1 \leq i \leq k$ and $1 \leq j \leq |\pi_i|)$ **do**
swap$(1, \phi^i_j)$ to get the path $s \rightarrow u_{i_j} = (\pi_i^{(j)} 1)\pi_1 \ldots \pi_{i-1}\pi_{i+1} \ldots \pi_k$;
else if $(l = 2)$ **then**
{Find paths:
$s \rightarrow (\pi_1 1)\pi_2 \ldots \pi_k \rightarrow u_{1_1} = (\phi^1_2 1)\pi_2 \ldots \pi_k$;
$s \rightarrow u_{1_2} = (\pi_1^{(2)} 1)\pi_2 \ldots \pi_k$;
$s \rightarrow u_{2_1} = (\pi_2 1)\pi_1\pi_3 \ldots \pi_k$;
$s \rightarrow (\pi_2^{(2)} 1)\pi_1\pi_3 \ldots \pi_k \rightarrow u_{2_2} = (\phi^2_1 1)\pi_1 \ldots \pi_k$;
for $(3 \leq i \leq k$ and $1 \leq j \leq |\pi_i|)$ **do**
swap$(1, \phi^i_j)$ to get the path $s \rightarrow u_{i_j} = (\pi_i^{(j)} 1)\pi_1 \ldots \pi_{i-1}\pi_{i+1} \ldots \pi_k$;
}
else /* $l \geq 3$. */
{**for** $(1 \leq i \leq l-1)$ **do** {Find paths:
$s \rightarrow (\pi_i 1)\pi_{i+1}\pi_1 \ldots \pi_{i-1}\pi_{i+2} \ldots \pi_k \rightarrow u_{i_1} = (\phi^i_2 1)\pi_{i+1} \ldots \pi_k$;
$s \rightarrow u_{i_2} = (\pi_i^{(2)} 1)\pi_1 \ldots \pi_{i-1}\pi_{i+1} \ldots \pi_k$; }
Find paths:
$s \rightarrow (\pi_l 1)\pi_1 \ldots \pi_{l-1}\pi_{l+1} \ldots \pi_k \rightarrow u_{l_1} = (\phi^l_2 1)\pi_1 \ldots \pi_k$;
$s \rightarrow u_{l_2} = (\pi_l^{(2)} 1)\pi_1 \ldots \pi_{l-1}\pi_{l+1} \ldots \pi_k$;
for $(l+1 \leq i \leq k$ and $1 \leq j \leq |\pi_i|)$ **do**
swap$(1, \phi^i_j)$ to get the path $s \rightarrow u_{i_j} = (\pi_i^{(j)} 1)\pi_1 \ldots \pi_{i-1}\pi_{i+1} \ldots \pi_k$;
}
for $(1 \leq i \leq k$ and $1 \leq j \leq |\pi_i|)$ **do** merge-then-reduce(u_{i_j});
2. **for** (each p_j with $p_j = j$) **do**
{swap$(1, p_j)$ to get $s \rightarrow u_j = (j1)\pi_1 \ldots \pi_k$; merge-then-reduce$(u_j)$; }
end.

Figure 4: 3-separated paths Algorithm for $|\pi_0| = 1$ in G_n

Algorithm II
Input: A node $s = \pi_0\pi_1 \ldots \pi_k$ with $|\pi_0| \geq 2$ in G_n.
Output: $(n-1)$ 3-separated paths from s to I.
begin
1. **if** $(|\pi_0| = 2)$ **then** find $s \rightarrow (1)\pi_1\pi_2 \ldots \pi_k \rightarrow u_2 = (\pi_1 1)\pi_2 \ldots \pi_k$
else swap(ϕ^0_1, ϕ^0_2) to get $s \rightarrow u_2 = (\phi^0_2 \ldots \phi^0_{|\pi_0|-1} 1)\pi_1 \ldots \pi_k$;
for $(3 \leq i \leq |\pi_0|)$ **do**
{swap(ϕ^0_1, ϕ^0_i) to find $s \rightarrow (\phi^0_i \ldots \phi^0_{|\pi_0|-1} 1)(\phi^0_1 \ldots \phi^0_{i-1})\pi_1 \ldots \pi_k$;
for $(i \leq j < |\pi_0|)$ **do** swap(ϕ^0_j, ϕ^0_{j+1}); /* reduce $(\phi^0_i \ldots \phi^0_{|\pi_0|})$ to (1) */
if $(i < |\pi_0|)$ **then** swap$(1, \phi^0_1)$ to find $s \rightarrow u_i = (\phi^0_1 \ldots \phi^0_{i-1} 1)\pi_1 \ldots \pi_k$
else {swap$(1, \phi^0_2)$ to find $s \rightarrow u_i = (\phi^0_2 \ldots \phi^0_{|\pi_0|-1}\phi^0_1 1)\pi_1 \ldots \pi_k$;
if $(|\pi_0| = 2)$ **then** find $s \rightarrow (\phi^0_2\phi^0_1 1)\pi_1 \ldots \pi_k \rightarrow u_3 = (\phi^0_1 1)\pi_1 \ldots \pi_k$;
}
for $(2 \leq i \leq |\pi_0|)$ **do** merge-then-reduce(u_i);
2. **for** $(1 \leq i \leq k$ and $1 \leq j \leq |\pi_i|)$ **do**
{swap(ϕ^0_1, ϕ^i_j); /* merge cycles π_0 and π_i into one cycle $(\pi_i^{(j)}\pi_0)$. */
if $(i = 1$ and $j = 1)$ **then** find $s \rightarrow u_{11} = (\pi_1\pi_0)\pi_2 \ldots \pi_k$
else {swap$(1, \phi^i_j)$ to find
$s \rightarrow (1)(\phi^i_j \ldots \phi^i_{|\pi_i|}\phi^i_1 \ldots \phi^i_{j-1}\phi^0_1 \ldots \phi^0_{|\pi_0|-1})\pi_1 \ldots \pi_{i-1}\pi_{i+1} \ldots \pi_k$;
if $(j < |\pi_j|)$ **then** swap$(1, \phi^i_{j+1})$ to find $s \rightarrow u_{ij} = (\phi^i_{j+1} \ldots \phi^i_{j-1}\phi^0_1 \ldots \phi^0_{|\pi_0|-1}\phi^i_j 1)\pi_1 \ldots \pi_k$
else swap$(1, \phi^i_1)$ to find $s \rightarrow u_{ij} = (\phi^i_1 \ldots \phi^i_{|\pi_i|-1}\phi^0_1 \ldots \phi^0_{|\pi_0|-1}\phi^i_{|\pi_i|} 1)\pi_1 \ldots \pi_k$;}
merge-then-reduce(u_{ij});
}
3. **for** (each p_j with $p_j = j$) **do**
{swap(j, ϕ^0_1), swap$(1, j)$, and swap$(1, \phi^0_1)$ to find $s \rightarrow u_j = (\phi^0_1 \ldots \phi^0_{|\pi_0|-1} j1)\pi_1 \ldots \pi_k$;
merge-then-reduce(u_j) };
end.

Figure 5: 3-separated paths Algorithm for $|\pi_0| \geq 2$ in G_n.