An Algorithm for k-pairwise Cluster-fault-tolerant Disjoint Paths in a Burnt Pancake Graph

Masato Tokuda, Yuki Hirai, and Keiichi Kaneko*
Graduate School of Engineering, Tokyo University of Agriculture and Technology, Koganei-shi, Tokyo, Japan
{50014646126@st,yhirai@cc,k1kaneko@cc}.tuat.ac.jp

Abstract— *In this paper, we focus on the pairwise cluster-fault-tolerant disjoint path routing problem in a burnt pancake graph, and propose an algorithm that solves the problem in a polynomial time of the degree of the graph. That is, in an n-burnt pancake graph with $n-2k+1$ faulty-cluster whose diameters are at most 3, the algorithm can construct fault-free disjoint paths between k pairs of nodes. The time complexity of the algorithm is $O(kn^3)$ and the maximum path length is $2n+15$.*

Keywords: pancake graph, multicomputer, interconnection network, parallel processing

1. Introduction

In the near future, processing performance of a sequential computer is expected to reach a ceiling because of limitations in technology. With this expectation, the field of parallel and distributed computation is taking on increasing importance, and studies on massively parallel computers are eagerly conducted recently. An interconnection network provides a topology to construct a massively parallel computer, and many topologies have been proposed and studied to interconnect many computers.

One of the factors that determine the performance of an interconnection network is fault tolerance. As the number of processors in a parallel computer increases, the probability of existence of faulty processors also increases. In practice, we face with the situation where not only the single processor fault but also a set of faulty nodes will arise. To address a fault-torelant routing problem has a merit to estabilish a fault-free communication, and there are many research activities about it. Similaly, to address a disjoint path routing problem has a merit to estabilish communication that gets no interference from other communication, and it is also studied very hard.

In this paper, we have focused on a burnt pancake graph [1], [2], [3], [4], [5], which is derived from a pancake graph [6], [7], [8], [9], [10], [11], [12] of a Cayley graph. A burnt pancake graph can connect many nodes with a small degree. Also, burnt pancake graphs are expected to fill in the gaps of incremental expandability of pancake graphs because they can connect different numbers of nodes from pancake graphs. However, there are many unsolved problem with a burnt pancake graph such as the shortest-path routing problem, the pairwise cluster-fault-tolerant routing problem, and so on.

In this paper, we pick up the pairwise cluster-fault-tolerant routing problem among the unsolved problems in a burnt pancake graph. For this problem, we propose an algorithm that solves it in a polynomial time of the degree of the burnt pancake graph. That is, in a n-burnt pancake graph with at most $n-2k+1$ faulty clusters whose diameters are at most 3, for k pairs of the source and destination nodes, we prove that our algorithm can construct k fault-free disjoint paths between them. We also prove that the time complexity of the algorithm is $O(kn^3)$ and the maximum path length is $2n+15$.

2. Preliminaries

In this section, we first introduce a definition of a burnt pancake graph and related definitions.

Definition *1:* A permutation $\boldsymbol{u} = (u_1, u_2, \ldots, u_n)$ that satisfies that $\{|u_1|, |u_2|, \ldots, |u_n|\} = \langle n \rangle$ is called a signed permutation where $\langle n \rangle = \{1, 2, \ldots, n\}$.

Definition *2:* For a signed permutation $\boldsymbol{u} = (u_1, u_2, \ldots, u_n)$ and an integer i $(1 \leq i \leq n)$, the signed prefix reversal opeartion $\boldsymbol{u}^{(i)}$ is defined by $\boldsymbol{u}^{(i)} = (-u_i, -u_{i-1}, \ldots, -u_2, -u_1, u_{i+1}, \ldots, u_n)$.

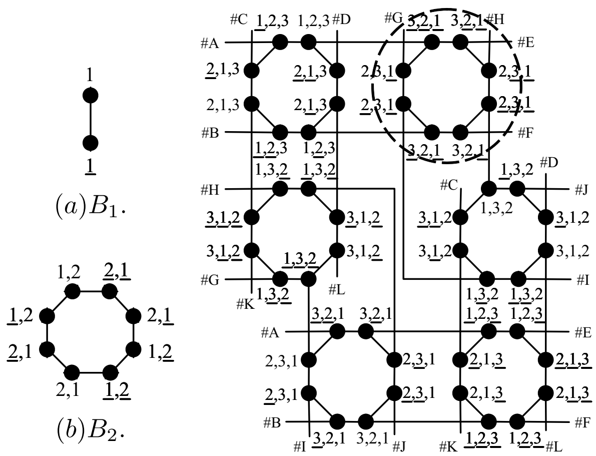
We use the notation $\boldsymbol{u}^{(i,\cdots,j,k)}$ as a short hand of a signed prefix reversal operation $\boldsymbol{u}^{(i,\cdots,j)^{(k)}}$. A signed prefix reversal operation is invertible and $\boldsymbol{u}^{(i,i)} = \boldsymbol{u}$ holds.

Definition *3:* If a graph $G(V, E)$ satisfies the conditions that $V = \{(u_1, u_2, \ldots, u_n) | (u_1, u_2, \ldots, u_n)$ is a signed permutation of $\langle n \rangle\}$ and $E = \{(\boldsymbol{u}, \boldsymbol{u}^{(i)}) | \boldsymbol{u} \in V, 1 \leq i \leq n\}$, G is called an n-burnt pancake graph.

In this paper, we denote B_n and $\underline{i}$ to represent an n-burnt pancake graph and $-i$, respectively.

A B_n is a symmetric graph, and the number of nodes, the number of edges, the degree, and the connectivity are $n! \times 2^n$, $n! \times n \times 2^{n-1}$, n, and n, respectively. There is no shortest-path routing algorithm found for a B_n in time

978-1-4673-9795-7/15 $31.00 © 2015 IEEE
DOI 10.1109/CSCI.2015.117



$(a)B_1$. $(b)B_2$. $(c)B_3$.

Fig. 1: Examples of burnt pancake graphs.

complexity of the polynomial order of n. However, the fact that $d(B_n) \leq 2n$ is proved.

Definition *4:* In a B_n, for an arbitrary node $\boldsymbol{u} = (u_1, u_2, \ldots, u_n)$ and an arbitrary integer k $(1 \leq |k| \leq n)$, an extended signed prefix reversal operation $\boldsymbol{u}^{([k])}$ is defined by

$$\boldsymbol{u}^{([k])} = \begin{cases} \boldsymbol{u} \to \boldsymbol{u}^{(i)} & \text{if } u_i = \underline{k}, \\ \boldsymbol{u} \to \boldsymbol{u}^{(i)} \to \boldsymbol{u}^{(i,1)} & \text{if } u_i = k. \end{cases}$$

Definition *5:* In a B_n, the sub graph induced by the subset of nodes that have k at the rightmost positions in their permutations is isomorphic to a B_{n-1}. The sub graph is specified by $B_{n-1}(k)$ by using the k as its index. A B_n is decomposable into $2n$ B_{n-1}'s that are mutually disjoint. Each sub graph is called a sub burnt pancake graph. In addition, the sub burnt pancake graph that contains the node $\boldsymbol{u}(\in B_n)$ is denoted by $B_{n-1}(\boldsymbol{u})$.

Definition *6:* A connected sub graph in a graph is called a cluster. If all of the nodes in a cluster are faulty, the cluster is called a faulty cluster. In addition, in a graph $G(V, E)$ the nodes defined by $\arg\min_{\boldsymbol{c} \in V} \sum_{\boldsymbol{v} \in V} d(\boldsymbol{c}, \boldsymbol{v})$ are called the centers of the graph G.

Definition *7:* In a B_n with faulty clusters, if a $B_{n-1}(k)$ does not include any center of the faulty clusters, it is called a candidate sub burnt pancake graph and denoted by $CB_{n-1}(k)$.

Definition *8:* The set that consists of the nodes that have $\underline{j}$ and i at the leftmost and rightmost positions in their permutations, respectively, is called a portset from $B_{n-1}(i)$ to $B_{n-1}(j)$, and denoted by $P(i, j)$.

Definition *9:* In a B_n, for an arbitrary node $\boldsymbol{u} = (u_1, u_2, \ldots, u_n)$, if u_i and u_{i+1} satisfies the following condition, u_i and u_{i+1} are called adjacent:

$$\boldsymbol{u}_{i+1} = \begin{cases} 1 & \text{if } u_i = n, \\ n & \text{if } u_i = \underline{1}, \\ u_i + 1 & \text{if } u_i \neq \underline{1}, n. \end{cases}$$

An element that is not adjacent to any other elements or elements that are successively adjacent are called a block. If a node consists of multiple blocks, two blocks can be put into one block by at most two signed prefix reversal operations. Therefore, in a B_n, routing between two arbitrary nodes can be reduced by at most two signed prefix reversal operations into routing in a B_{n-1}.

Theorem *1:* In a B_n, for two non-faulty nodes $\boldsymbol{s} = (s_1, s_2, \ldots, s_n)$, $\boldsymbol{t} = (t_1, t_2, \ldots, t_n)$ and a set of faulty nodes F $(|F| \leq n-1)$, we can construct a fault-free path between $\boldsymbol{s}$ and $\boldsymbol{t}$ of length at most $2n+4$ in time complexity $O(n^2)$.

3. Algorithm

In this section, we describe an algorithm that solves the pairwise cluster-fault-tolerant disjoint paths problem in a burnt pancake graph.

3.1 Lemmas

Lemma *1:* For two distinct nodes $\boldsymbol{u}$ and $\boldsymbol{v}$ in a port set $P(l, m)$ $(1 \leq |l|, |m| \leq n,\ |l| \neq |m|)$, the distance between them $L(\boldsymbol{u}, \boldsymbol{v})$ is no less than 3.

Lemma *2:* There is not a cycle in a B_n whose length is less than 8.

Lemma *3:* In a B_n, if there are at most $n-2k+1$ faulty clusters whose diameters are at most 3, there are at least $4k-2$ candidate sub burnt pancake graphs.
(Proof) The centers of a faulty cluster of diameter 3 exist at most 2 sub burnt pancake graphs. If there are $n-2k+1$ faulty clusters, the centers of the faulty clusters exist at most $2n-4k+2$ sub burnt pancake graphs. Therefore, because there are $2n$ sub burnt pancake graphs in a B_n, there are at least $4k-2$ candidate sub burnt pancake graphs.

Lemma *4:* In a B_n, for a node $\boldsymbol{u} = (u_1, u_2, \ldots, u_n)$, we can construct n disjoint paths of length at most 3 from $\boldsymbol{u}$ to n distinct sub burnt pancake graphs $B_{n-1}(k)$ $(k \neq |u_n|)$ in $O(n^2)$ time complexity.
(Proof) We can construct the following paths:
$\boldsymbol{u} \to \boldsymbol{u}^{(n)} \in B_{n-1}(\underline{u_1})$ $(i = n)$,
$\boldsymbol{u} \to \boldsymbol{u}^{(i)} \to \boldsymbol{u}^{(i,n)} \in B_{n-1}(u_i)$ $(1 \leq i \leq n-1)$.
Among the latter paths, for $n-2$ paths except for $B_{n-1}(u_1)$, we can construct paths as follows:
$\boldsymbol{u} \to \boldsymbol{u}^{(i)} \to \boldsymbol{u}^{(i,1)} \to \boldsymbol{u}^{(i,1,n)} \in B_{n-1}(\underline{u_i})$ $(2 \leq i \leq n-1)$.
Hence, for n of $2n-2$ $B_{n-1}(k)$ except for $B_{n-1}(u_n)$ and

$B_{n-1}(\underline{u}_n)$, we can construct disjoint paths except for $\boldsymbol{u}$. It takes $O(n)$ time to construct one path, it takes $O(n^2)$ time in total.

Figure 2 shows n disjoint paths from a node $\boldsymbol{u}$ to n distinct sub burnt pancake graphs.

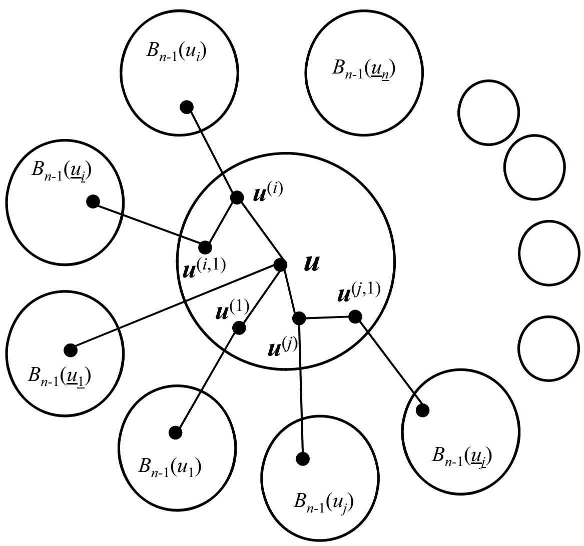


Fig. 2: n disjoint paths from a node $\boldsymbol{u}$ to n distinct sub burnt pancake graphs constructed in Lemma 4.

Lemma *5:* In a B_n, for a non-faulty node $\boldsymbol{u} = (u_1, u_2, \ldots, u_n)$ and n candidate sub burnt pancake graphs $CB_{n-1}(k)$ $(k \neq |u_n|)$, a faulty cluster whose diameter is at most 3 can overlap at most one of the n paths of length at most 3 that are given in Lemma 4.

(Proof) Assume that there is a faulty cluster that blocks two among the paths given by Lemma 4. Assume that the blocked paths are $\boldsymbol{u} \to \boldsymbol{u}^{(i)} \to \boldsymbol{u}^{(i,1)} \to \boldsymbol{u}^{(i,1,n)}$ and $\boldsymbol{u} \to \boldsymbol{u}^{(j)} \to \boldsymbol{u}^{(j,1)} \to \boldsymbol{u}^{(j,1,n)}$ $(2 \leq i \leq n-1,$ $2 \leq j \leq n-1)$. From Lemma 2, there is no faulty cluster that blocks $\boldsymbol{u}^{(i,1)}$ and $\boldsymbol{u}^{(j,1)}$ simultaneously. For an arbitrary pair of paths among the paths given by Lemma 4, similar discussion holds.

Lemma *6:* In a B_n, for a node $\boldsymbol{u} = (u_1, u_2, \ldots, u_n)$ and a sub burnt pancake graph $B_{n-1}(k)$, $(k \neq u_1, |u_n|)$, we can construct n disjoint paths of length at most 5 from $\boldsymbol{u}$ to the $B_{n-1}(k)$ in $O(n^2)$ time complexity.

(Proof) We cosider three cases depending on k.

Case 1 $|k| \neq |u_1|, |u_n|$

We can construct n paths as follows.

$\boldsymbol{u} \to \boldsymbol{u}^{(i)} \rightsquigarrow \boldsymbol{u}^{(i,[k])} \to \boldsymbol{u}^{(i,[k],n)}$ $(1 \leq i \leq n,\ k \neq u_i)$

$\boldsymbol{u} \to \boldsymbol{u}^{(i)} \to \boldsymbol{u}^{(i,n)}$ $(k = u_i)$

$\boldsymbol{u} \to \boldsymbol{u}^{(i)} \to \boldsymbol{u}^{(i,1)} \to \boldsymbol{u}^{(i,1,n)}$ $(\underline{k} = u_i)$

Then, the lengths of these n paths are at most 4.

Case 2 $k = \underline{u}_n$

We can construct n paths as follows.

$\boldsymbol{u} \to \boldsymbol{u}^{(i)} \to \boldsymbol{u}^{(i,n)} \to \boldsymbol{u}^{(i,n,1)} \to \boldsymbol{u}^{(i,n,1,n)}$ $(1 \leq i \leq n-1)$

$\boldsymbol{u} \to \boldsymbol{u}^{(n)} \to \boldsymbol{u}^{(n,1)} \to \boldsymbol{u}^{(n,1,n)}$ $(i = n)$

Then, among n paths, there is one path whose length is 3 and there are $n-1$ paths whose lengths are 4.

Case 3 $k = u_1$

We can construct n paths as follows.

$\boldsymbol{u} \to \boldsymbol{u}^{(1)} \to \boldsymbol{u}^{(1,n)}$ $(i = 1)$

$\boldsymbol{u} \to \boldsymbol{u}^{(i)} \to \boldsymbol{u}^{(i,1,i)} \to \boldsymbol{u}^{(i,1,i,1)} \to \boldsymbol{u}^{(i,1,i,1,i)}$ $(2 \leq i \leq n)$

Then, among n paths, there is one path whose length is 1 and there are $n-1$ whose lengths are 5.

From above, we can construct n paths of length at most 5 from $\boldsymbol{u}$ to $B_{n-1}(k)$ that are disjoint except for $\boldsymbol{u}$. Since it takes $O(n)$ time to construct one path, it takes $O(n^2)$ time to construct n paths.

Lemma *7:* In a B_n with at most $n-2$ faulty nodes, from arbitrary nodes $\boldsymbol{u}_{l_i}$ in $n-1$ distinct $CB_{n-1}(l_i)$ $(1 \leq |i_i| \leq n,$ $1 \leq i \leq n$ where $|l_j| \neq |l_i|$ if $i \neq j)$, we can construct $n-1$ disjoint fault-free paths of length at most 7 via $CB_{n-1}(p)$ $(|p| \neq \forall |l_i|)$.

(Proof) For each $\boldsymbol{u}_{l_i}$, let us consider n paths of lengths at most 5 to $CB_{n-1}(p)$ given by Lemma 6. Then, there is at least one fault-fee path $\boldsymbol{u}_{l_i} \rightsquigarrow \boldsymbol{v}_{l_i} \in P(p, l_i)$ among the paths that do not include $\boldsymbol{u}_{l_i}^{(n)}$. Then, we can construct a path $\boldsymbol{v}_{l_i} \to \boldsymbol{v}_{l_i}^{(1)} \in P(p, \underline{l}_i) \to \boldsymbol{v}_{l_i}^{(1,n)} \in CB_{n-1}(\underline{l}_i)$. Since port sets are mutually disjoint, the $n-1$ paths are disjoint. Then, the lengths of the constructed paths are at most 7.

Lemma *8:* In a B_n with $n-2$ faulty nodes, we can construct $n-1$ fault-free disjoint paths of length at most 7 from $n-1$ non-faulty nodes $\boldsymbol{u}_{l_i}$ in distinct $n-1$ $CB_{n-1}(l_i)$ $(1 \leq |l_i| \leq n$ $1 \leq i \leq n$ where $|l_i| \neq |l_j|$ if $i \neq j)$ to $CB_{n-1}(\underline{l}_i)$ via a $CB_{n-1}(p)$ $(|p| \neq \forall |l_i|)$.

Figure 3 shows the $n-1$ fault-free disjoint paths of length at most 7 constructed in Lemma 8.

3.2 Algorithm

In this section, we show an algorithm for a cluster-fault-tolerant k-pairwise disjoint path routing, and estimate the maximum path length and its time complexity. In a B_n, for k pairs of source and destination nodes $(3 \leq k \leq \lceil n/2 \rceil)$, the algorithm first constructs paths $\boldsymbol{s}_i \rightsquigarrow \boldsymbol{s}'_i$ and $\boldsymbol{t}_i \rightsquigarrow \boldsymbol{t}'_i$ where $\boldsymbol{s}'_i$ and $\boldsymbol{t}'_i$ belong to a same B_{n-1} and the paths do not include any node on the other paths $\boldsymbol{s}_j \rightsquigarrow \boldsymbol{s}'_j$ nor $\boldsymbol{t}_j \rightsquigarrow \boldsymbol{t}'_j$ where $j \neq i$. Then, it connects $\boldsymbol{s}'_i$ and $\boldsymbol{t}'_i$ by a fault-free path in the B_{n-1} by using the fault-tolerant routing algorithm. The B_{n-1} is called the target sub burnt pancake graph for the pair of nodes $\boldsymbol{s}_i$ and $\boldsymbol{t}_i$, and denoted by $B_{n-1}(l_i)$ $(1 \leq |l_i| \leq n,$ $1 \leq i \leq k)$. Also, in the rest of this paper, we assume thta the candidate sub burnt pancake graph for $\boldsymbol{s}_i$ satisfies the

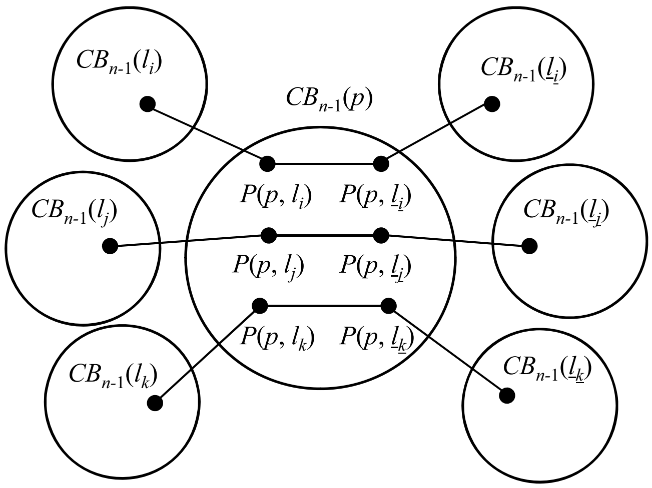


Fig. 3: $n-1$ fault-free disjoint paths of length at most 7 constructed in Lemma 8.

condition that it does not include any nodes on the other paths $\boldsymbol{s}_j \rightsquigarrow \boldsymbol{s}'_j$ nor $\boldsymbol{t}_j \rightsquigarrow \boldsymbol{t}'_j$ in addition to the condition that it does not include any center of faulty clusters.

The algorithm consists of the following four steps.

Step 1) If there is a $B_{n-1}(m)$ that contains $\boldsymbol{s}_i$ or $\boldsymbol{t}_i$ and the $B_{n-1}(m)$ is a candidate sub burnt pancake graph for $\boldsymbol{s}_i$ or $\boldsymbol{t}_i$, assign the $B_{n-1}(m)$ to the target sub burnt pancake graph $B_{n-1}(l_i)$. If $\boldsymbol{s}_i$ and $\boldsymbol{t}_i$ are included in distinct $B_{n-1}(p)$ and $B_{n-1}(q)$, respectively, and both of $B_{n-1}(p)$ and $B_{n-1}(q)$ satisfy the conditions of candidate sub burnt pancakes for $\boldsymbol{s}_i$ and $\boldsymbol{t}_i$, either of them are assigned to $B_{n-1}(l_i)$. We can assign a target sub burnt pancake graph for each pair of source and destination nodes in $O(n)$ time.

Step 2) For each pair of the source node $\boldsymbol{s}_i = (s_{i1}, s_{i2}, \ldots, s_{in})$ and the destination nodes $\boldsymbol{t}_i$ to which any target sub burnt pancake graph is not assigned, construct a path from either of the source or destination nodes to a candidate sub burnt pancake graph for it. Here, we assume that we found a candidate sub burnt pancake graph for $\boldsymbol{s}_i$. Then, we can assign a target sub burnt pancake graph and construct a path by the following three sub steps.

Sub Step 2a) If the $B_{n-1}(\underline{s}_{i1})$ is a candidate sub burnt pancake graph for $\boldsymbol{s}_i$, we assign $B_{n-1}(\underline{s}_{i1})$ to the target sub burnt pancake graph $B_{n-1}(l_i)$, construct a path $\boldsymbol{s}_i \rightsquigarrow \boldsymbol{s}_i^{(n)}$ of length 1 to the sub burnt pancake graph, and let $\boldsymbol{s}_i^{(n)} = \boldsymbol{s}'_i$.

Sub Step 2b) If the $B_{n-1}(s_{ip})$ $(1 \leq p < n)$ is a candidate sub burnt pancake graph for $\boldsymbol{s}_i$, we try to construct a path $\boldsymbol{s}_i \rightsquigarrow \boldsymbol{s}_i^{(p)} \rightsquigarrow \boldsymbol{s}_i^{(p,n)}$ of length 2. If this path is fault-free and disjoint from other paths, we can assign $B_{n-1}(s_{ip})$ to the target sub burnt pancake graph $B_{n-1}(l_i)$, and let $\boldsymbol{s}_i^{(p,n)} = \boldsymbol{s}'_i$.

Sub Step 2c) If the $B_{n-1}(\underline{s}_{ip})$ $(1 < p < n)$ is a candidate sub burnt pancake graph for $\boldsymbol{s}_i$, we try to construct a path $\boldsymbol{s}_i \rightsquigarrow \boldsymbol{s}_i^{(p)} \rightsquigarrow \boldsymbol{s}_i^{(p,1)} \rightsquigarrow \boldsymbol{s}_i^{(p,1,n)}$ of length 3. If this path is fault-free and disjoint from other paths, we can assign $B_{n-1}(\underline{s}_{ip})$ to the target sub burnt pancake graph $B_{n-1}(l_i)$, and let $\boldsymbol{s}_i^{(p,1,n)} = \boldsymbol{s}'_i$.

In Step 2, we can assign a target sub burnt pancake graph for each pair of source and destination nodes by constructing a path of length at most 3 in $O(n^3)$ time.

Step 3) By Steps 1 and 2, for k pairs of nodes $\boldsymbol{s}_i$ and $\boldsymbol{t}_i$, target sub burnt pancake graphs $B_{n-1}(l_i)$ are assigned and at least one path from either of the nodes is constructed. Here, we construct a path to $B_{n-1}(l_i)$ from either of $\boldsymbol{s}_i$ or $\boldsymbol{t}_i$ from which a path to $B_{n-1}(l_i)$ has not been constructed. For simplicity, we assume that a path from $\boldsymbol{s}_i$ to $B_{n-1}(l_i)$ has been already contrcuted, and a path from $\boldsymbol{t}_i$ has not been constructed without loss of generality. Here, if $\underline{t}_{i1} = l_i$, we can construct a path $\boldsymbol{t}_i \to \boldsymbol{t}_j^{(n)}$ of length 1. If $\underline{t}_{in} \neq l_i$, we first check the n paths of lengths at most 4 given by Lemma 6. Among the $n-1$ paths of them that do not pass $\boldsymbol{t}^{(n)}$, if there is a path that is fault-free and disjoint from other paths, let the path be $\boldsymbol{t}_i \rightsquigarrow \boldsymbol{t}'_i$. If $t_{i1} = l_i$, we check the path $\boldsymbol{t}_j \to \boldsymbol{t}_j^{(1)} \to \boldsymbol{t}_j^{(1,n)}$ of lenfth 2. If it is fault-free and disjoint from other paths, let the path be $\boldsymbol{t}_j \rightsquigarrow \boldsymbol{t}'_j$. If there is not such path, or if $\underline{t}_{in} = l_i$, we construct a path to a candidate sub burnt pancake graph for $\boldsymbol{t}_i$ as similar to the Sub Steps 2a), 2b), and 2c). Let this candidate sub burnt pancake graph be $B_{n-1}(l'_i)$.

Then, this step is divided into two cases depending on l_i and l'_i to construct the path.

Case 1)$(l_i = \underline{l}'_i)$ For pairs of the nodes such that $l_i = \underline{l}'_i$ hold, we can construct disjoint paths of lengths at most 7 that pass a candidate sub burnt pancake graph $B_{n-1}(p)$ that does not include any source nor destination node from Lemma 7.

Case 2)$(l_i \neq \underline{l}'_i)$ If $l_i \neq \underline{l}'_i$, there is a fault-free path among the paths from $B_{n-1}(\underline{l}'_i)$ to $B_{n-1}(l_i)$ of length at most 5 given by Lemma 6.

In this step, we can construct a path of length at most 10 between a pair of a source node and a destination node in $O(n^3)$ time.

Step 4) For k pairs of nodes $\boldsymbol{s}_i$ and $\boldsymbol{t}_i$ $(1 \leq i \leq k)$, from Steps 1 to 3, we have constructed paths $\boldsymbol{s}_i \rightsquigarrow \boldsymbol{s}'_i(\in B_{n-1}(l_i))$ and $\boldsymbol{t}_i \rightsquigarrow \boldsymbol{t}'_i(\in B_{n-1}(l_i))$ where $B_{n-1}(l_i)$ is the target sub burnt pancake graph for $\boldsymbol{s}_i$ and $\boldsymbol{t}_i$. $B_{n-1}(l_i)$ does not include any node on $\boldsymbol{s}_j \rightsquigarrow \boldsymbol{s}'_j$ or $\boldsymbol{t}_j \rightsquigarrow \boldsymbol{t}'_j$ $(j \neq i)$, and contains at most $n-2k+1$ faulty nodes. Therefore, from Theorem 1, we can construct a path $\boldsymbol{s}'_i \rightsquigarrow \boldsymbol{t}'_i$ of length at most $2n+2$ in $O(n^2)$ time.

Consequently, our algorithm can construct each path $\boldsymbol{s}_i \rightsquigarrow \boldsymbol{t}_i$ of length at most $2n+15$ in $O(n^3)$ time. Therefore, it takes $O(kn^3)$ time to construct k paths.

4. Evaluation

To evaluate performance of our algorithm, we conducted a computer experiment. The algorithm constructed k disjoint fault-free paths between the k pairs of source and destination nodes in a n-burnt pancake graph with $n-2k+1$ faulty clusters whose diameters are 3. In this section, we give the method, the results, and consideration.

4.1 Method

In the experiment, we applied our algorithm to solve the k-pairwise disjoint cluster-fault-free paths problem ($3 \le k \le \lceil n/2 \rceil$) in a B_n ($5 \le n \le 40$). We repeated the following steps for 10,000 times and measured the average execution time and the maximum path length as well as the average path length.

1) We first set up $n-2k+1$ disjoint faulty clusters whose diameter is fixed to 3.
2) Then we select k source nodes $\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_k$ and k destination nodes $\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_k$ among non-faulty nodes.
3) We apply the algorithm to construct k disjoint fault-free paths $\boldsymbol{s}_i \rightsquigarrow \boldsymbol{t}_i$ ($1 \le i \le k$) and measure the execution time, the maximum path length, and the average path length.

Note that, in Step 4, we need a routing algorithm to construct a path between two arbitrary nodes in a non-faulty burnt pancake graph B_n. We have adopted the algorithm whose time complexity is $O(n^2)$ and the maximum path length is $2n$ by Cohen and Blum [1]. Therefore, the theoretical maximum path length by our algorithm becomes $2n + 15$.

4.2 Results

Figure 4 shows the results of the maximum path lengths and the average path lengths for $5 \le n \le 40$. In addition, Figure 5 shows the result of the average execution time for $5 \le n \le 40$ and $3 \le k \le \lceil n/2 \rceil$. From Figure 4, we can see that there is no path whose length attained the theoretical maximum path lengths. From Figure 5, the average execution time seems to converge to $O(n^{2.2})$.

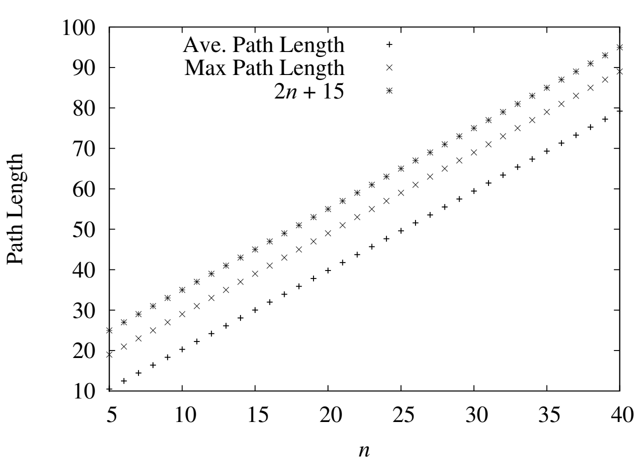


Fig. 4: Maximum and average path lengths of our algorithm

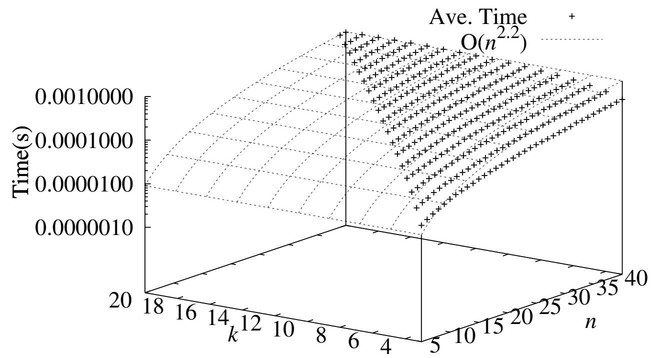


Fig. 5: Execution time of our algorithm

Acknowledgments

This study is partly supported by a Grant-in-Aid for Scientific Research (C) of the Japan Society for the Promotion of Science (JSPS) under Grant No. 25330079.

References

[1] D. S. Cohen and M. Blum, "On the problem of sorting burnt pancakes," *Discrete Applied Mathematics*, vol. 61, no. 2, pp. 105–120, 1995.

[2] T. Iwasaki and K. Kaneko, "Fault-tolerant routing in burnt pancake graphs," *Information Processing Letters*, vol. 110, no. 14-15, pp. 535–538, Jul. 2010.

[3] K. Kaneko, "An algorithm for node-to-set disjoint paths problem in burnt pancake graphs," *IEICE Transactions on Information and Systems*, vol. E86-D, no. 12, pp. 2588–2594, Dec. 2003.

[4] ——, "Hamiltonian cycles and hamiltonian paths in faulty burnt pancake graphs," *IEICE Transactions on Information and Systems*, vol. E90-D, no. 4, pp. 716–721, Apr. 2007.

[5] K. Kaneko and N. Sawada, "An algorithm for node-to-node disjoint paths problem in burnt pancake graphs," *IEICE Transactions on Information and Systems*, vol. E90-D, no. 1, pp. 306–313, Jan. 2007.

[6] S. B. Akers and B. Krishnamurthy, "A group-theoretic model for symmetric interconnection networks," *IEEE Transactions on Computers*, vol. 38, no. 4, pp. 555–566, Apr. 1989.

[7] D. W. Bass and I. H. Sudborough, "Pancake problems with restricted prefix reversals and some corresponding cayley networks," *Journal of Parallel and Distributed Computing*, vol. 63, no. 3, pp. 327–336, 2003.

[8] W. H. Gates and C. H. Papadimitriou, "Bounds for sorting by prefix reversal," *Discrete Mathematics*, vol. 27, pp. 47–57, 1979.

[9] M. H. Heydari and I. H. Sudborough, "On the diameter of the pancake network," *J. Algorithms*, vol. 25, no. 1, pp. 67–94, 1997.

[10] K. Kaneko and Y. Suzuki, "Node-to-set disjoint paths problem in pancake graphs," *IEICE Transactions on Information and Systems*, vol. E86-D, no. 9, pp. 1628–1633, Sep. 2003.

[11] K. Qiu, H. Meijer, and S. G. Akl, "Parallel routing and sorting on the pancake network," in *Proceedings of International Conference on Computing and Information*, ser. Lecture Notes in Computer Science, vol. 497. Springer Verlag, 1991, pp. 360–371.

[12] Y. Suzuki and K. Kaneko, "An algorithm for node-disjoint paths in pancake graphs," *IEICE Transactions on Information & Systems*, vol. E86-D, no. 3, pp. 610–615, Mar. 2003.