

# 特別研究計画

鄭 信雨

学籍番号 11834304

2017 年 1 月 21 日

## 目次

1	はじめに	3
2	目的	3
3	背景	3
4	様々なケイリーグラフ	4
4.1	ハイパーキューブ (hyper cube) $Q_n$	4
4.2	スターグラフ (star graph) $S_n$	4
5	スターグラフ	4
5.1	スターグラフの定義	4
5.2	スターグラフの性質	5
5.3	スターグラフ関連研究紹介	6
6	バブルソートグラフ	7
7	トランスポジショングラフ	7
7.1	トランスポジショングラフの定義	7
7.2	トランスポジショングラフの性質	7
7.3	トランスポジショングラフの関連研究紹介	8
8	パンケーキグラフ	8
8.1	パンケーキグラフの定義	9
8.2	パンケーキグラフの性質	9
8.3	パンケーキグラフの関連研究紹介	9
9	焦げたパンケーキグラフ	10
9.1	焦げたパンケーキグラフ	11
9.2	焦げたパンケーキグラフの性質	11
9.3	焦げたパンケーキグラフの関連研究紹介	11
10	まとめ	13
11	おわりに	13

## 1 はじめに

このレポートは特別研究計画最終レポートである。

## 2 目的

このレポートは今まで自身の研究のために調査してきた研究を簡潔にまとめ報告することを目的とする。

## 3 背景

逐次計算システムの性能向上の頭うちにより近年並列分散計算システムが様々な分野で活用されるようになってきた。それと同時に並列分散処理計算システムの研究開発も活発に行われている [1]。日本国内でも 2012 年 11 月スーパーコンピューター京が正式運用を開始したが、その僅か 4 年後の 2016 年 11 月には京の浮動小数演算性能の約 2.2 倍高性能な Oakforest-PACS スーパーコンピュータシステムが正式運用を開始するなど、並列分散処理計算機システムの活発な研究開発を確認することができる [2][3]。

並列分散性能に直接関連するのは各ノードの個別性能も重要であるが、ノード間の通信方式を決める位相もとても重要である。例えば任意の二つのノード間で情報交換が必要になった場合なるべく短い経路で情報交換ができることが望ましい。最も距離を短くするには全てのノードを隣接させることで可能にはなるが、数十万、数百万のノード間に全てリンクを設置するの現実的に不可能である。また、並列分散計算システムでは耐故障性の確保もとても重要である。巨大な数のノードを扱うので運用上一部のノードの故障を避けることはできない [4]。しかし、並列分散計算システムではたとえ一部のノードが故障していても可能な限り正しく動作する必要がある。この問題を解決するためにグラフ理論が用いられるようになった。並列分散システムの各ノードを頂点としリンクを辺にすることでグラフ理論を用いることができる。このような背景より並列分散システムの位相を与えることができる様々な相互結合網が提案されてきた。例えば、メッシュ や ハイパーキューブ、パンケーキなどがある。そして、それらの相互結合網において耐故障経路選択解法や並列通信経路解法などが数多く提案されてきた。本レポートでは相互結合網における耐故障性および並列通信経路の解法である頂点間、頂点と頂点集合間、頂点集合間、素な経路選択アルゴリズムに関する論文に関して解説する。

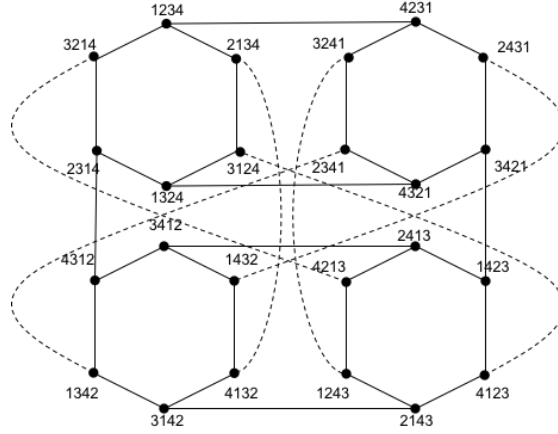


図 1 4 - Star graph

## 4 様々なケイリーグラフ

本章では本レポートで取り扱う様々なケイリーグラフに関して述べる.

### 4.1 ハイパーキューブ (hyper cube) $Q_n$

### 4.2 スターグラフ (star graph) $S_n$

$s_1 s_2 \dots s_n$  を 1 から  $n$  までの  $n$  種類の記号で作られる順列とする.  $SWAP_j(s_1 s_2 \dots s_n) = s_j s_2 \dots s_{j-1} s_1 s_{j+1} \dots s_n$  と定義する. 無向グラフ  $G(V, E)$  に対して  $n$ -star graph  $S_n = (V, E)$  の  $V, E$  を以下に示す.

$$V = \{(u_1 u_2 \dots u_n) | (u_1 u_2 \dots u_n) \text{ は } 1, 2, \dots, n \text{ の順列} \}$$

$$E = \{(u, SWAP_i(u)) | u \in V, 2 \leq i \leq n\}$$

図 4.2 に  $n$  が 4 の場合のスターグラフを示す。

## 5 スターグラフ

この節では、スターグラフに関する主な性質および有用な経路選択アルゴリズムに関する論文を紹介する。

### 5.1 スターグラフの定義

$s_1 s_2 \dots s_n$  を 1 から  $n$  までの  $n$  種類の記号で作られる順列とする.  $SWAP_j(s_1 s_2 \dots s_n) = s_j s_2 \dots s_{j-1} s_1 s_{j+1} \dots s_n$  と定義する. 無向グラフ  $G(V, E)$  に対して  $n$ -star graph  $S_n = (V, E)$

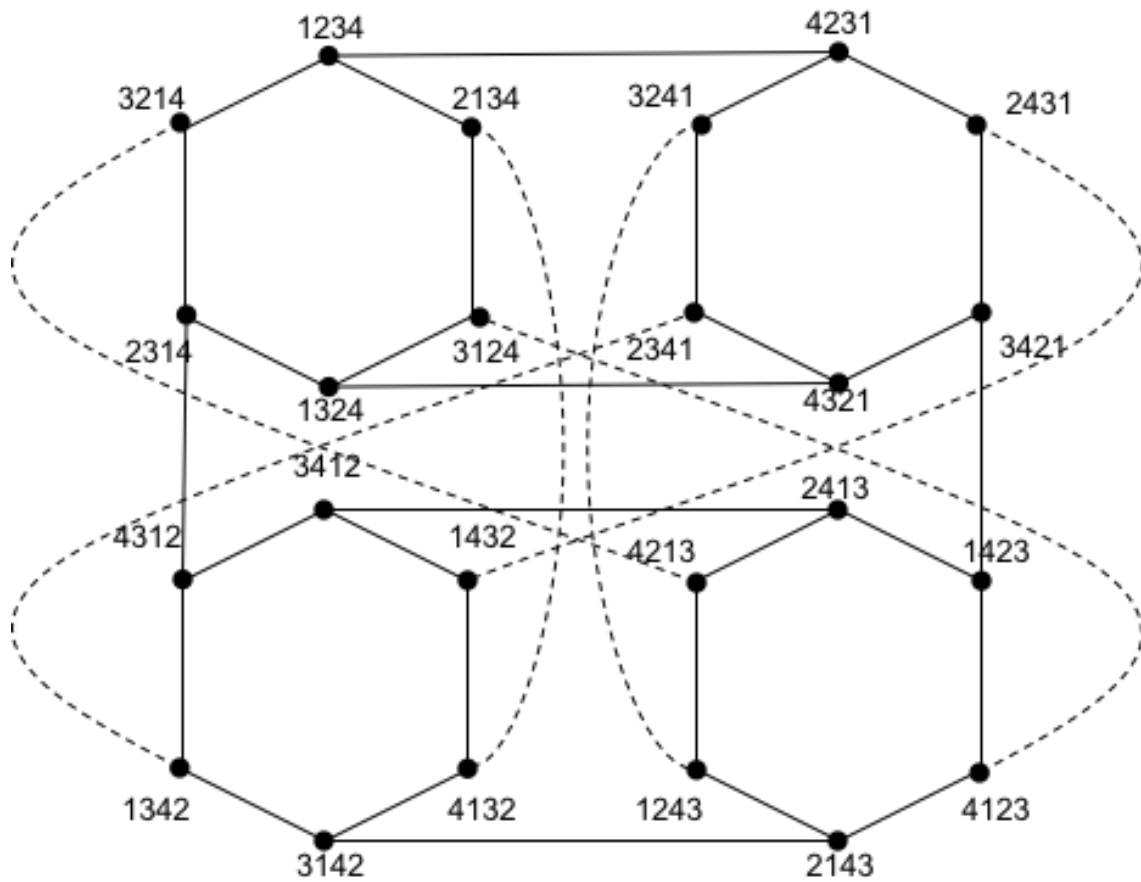


図2 3,4 - Star graph

は  $V = \{(u_1 u_2 \dots u_n) | (u_1 u_2 \dots u_n) \text{ は } 1, 2, \dots, n \text{ の順列}\}$  と  $E = \{(u, SWAP_i(u)) | u \in V, 2 \leq i \leq n\}$  である。

## 5.2 スターグラフの性質

スターグラフは再帰性と対称性をグラフでハイパーキューブに代わる位相として注目を集めている。表1にスターグラフの性質を示す。

表1  $n$ -スターグラフの性質

頂点数	次数	連結度	直径
$n!$	$n - 1$	$n - 1$	$\lfloor 3(n - 1)/2 \rfloor$

## 5.3 スターグラフ関連研究紹介

### 5.3.1 Selection, Routing, and Sorting on the Star Graph

この節ではスターグラフにおける Selection、Routing、Sorting に関してより効率的な解法を論文 [5] に関して述べる。

Sorting は与えられた順列を昇順、もしくは降順に並べかえる工程である。

Routing はパケット情報を出発地点より目的地点まで送ることである。

この論文ではまず記号列に対する Prefix computation を  $O(n^2)$  で達成する解法を提案する。実際にスターグラフにおける Prefix computation の最適な時間計算量は  $O(n \log n)$  ですでに提案されている [6] が本論文で提案された解法は並列処理が可能になるように工夫されているため、 $n$  個の記号列に対する Prefix computation も  $O(n^2)$  で可能であることを提案している。この Prefix computation の解法を彼らの他の解法のサブルーチンとして活用している。この解法を用いることで  $n$ -スターグラフでの Sorting を  $O(n^3)$  で実現している。また、Selection に関しては  $O(n^2)$  で Routing に関しては  $O(n^3)$  の解法を提案している。特に Routing に関しては既知の解法は  $O(n^3 \log n)$  であること比べ改善されている。これは彼らが提案した Prefix computation の解法が並列処理をした場合  $O(n^2)$  で解決できるため、既存の  $O(n^2 \log n)$  より改善されたためである。

### 5.3.2 Node-to-node cluster fault tolerant routing in star graphs

この節ではスターグラフにおける頂点間耐クラスタ故障ルーティングアルゴリズムに関する研究 [7] を紹介する。ここでクラスタ故障とは、故障頂点の集合である。 $n$ -スターグラフは  $n-1$  の次数を持つ正則グラフであるため、 $n-1$  の連結度を持つ。メンガーの定理により  $n$ -グラフでは高々  $n-2$  子の任意の故障頂点に耐えることができるがこの論文ではさらに直径が高々 2 である高々  $n-2$  のクラスタ故障に対して非故障出発頂点と目的頂点間に経路を構築する手法を提案している。この提案手法は最大経路長が  $\lfloor 3(n-1)/2 \rfloor + 8$  である場合計算量が  $O(|\text{クラスタ}| + n)$  であり、最大経路長が  $\lfloor 3(n-1)/2 \rfloor + 6$  である場合計算量が  $O(n^2)$  である。提案手法のキーポイントは  $n$ -スターグラフが  $n$  個の  $n-1$ -スターグラフを含んでいる再帰的な構造を利用している。論文では、頂点順列の最右記号  $k (1 \leq k \leq n)$  が同じである順列より誘発される  $n-1$ -サブスターグラフにを  $k$ -サブグラフとする。そして、 $i$ -サブグラフ内部より長さが高々 2 以下で  $j$ -サブグラフに移動する頂点集合を  $i$  より  $j$  へのポート集合として定義している。このようなポート集合は任意の  $k$ -サブグラフ内部に  $n-1$  個存在する。本論文ではこのポート集合を活用し頂点間耐クラスタ故障ルーティング手法を提案している。

### 5.3.3 Node-to-set disjoint paths problem in star graphs

この節ではスターグラフにおける頂点と頂点集合間の素な経路選択問題に関する二つの解法を提案した論文 [9] に関して述べる。

スターグラフにおける頂点と頂点集合間の素な経路選択問題は [8] で最初に解法が提案された。[8]

で提案された手法は  $n$ -スターグラフ内の  $n - 1$  の素な経路を構築するのに  $O(n^2)$  の時間計算量を必要とし、経路長は最大  $5(n - 2)$  であった。これに対し本論文で提案された手法では時間計算量は同じであるが、最大経路長が  $\lfloor 3(n - 1)/2 \rfloor + 3$  である *simple algorithm* と  $\lfloor 3(n - 1)/2 \rfloor + 2$  である *refined simple algorithm* を提案した。*simple algorithm* は大きく二つのサブルーチン *SCATTER* と *MERGE-THEN-REDUCE* で構成されている。*SCATTER* では与えられた順列組に対して記号 1 を先頭に持ってくる作業を行う。この作業で得られた結果はすべて素な経路になる。例えば *SCATTER*([13241], [4321], [3421]) により次の三つの素な経路が構築される。 $P_1 : v_1 = t_l = [1324], P_2 : [4321] \rightarrow [2341] \rightarrow v_2 = [1342], P_3 : [3421] \rightarrow v_3 = [1423]$ .

$n$ -スターグラフ内の複数の頂点ペア間の最短経路はお互いに素でない可能性がある。しかし、本論文で提案されている二つ目のサブルーチン *MERGE-THEN-REDUCE* では一定の条件を満たす頂点集合と出発頂点間で互いに素な最短経路を構築することができる。この *MERGE-THEN-REDUCE* が *simple algorithm* で構築された  $n - 1$  個の経路が互いに素であるための重要な役割を果たしている。

最後に *simple algorithm* より最大経路長が 1 少ない *refined simple algorithm* は *simple algorithm* で最初に行われる *SCATTER* 操作時に検知される最悪のケースに対して再度経路を改善することを実装している。

## 6 バブルソートグラフ

## 7 トランスポジショングラフ

この節では、トランスポジショングラフに関する主な性質および有用な経路選択アルゴリズムに関する研究を紹介する。

### 7.1 トランスポジショングラフの定義

$s_1 s_2 \dots s_n$  を 1 から  $n$  までの  $n$  種類の記号で作られる順列とする。 $TP_{(i,j)}(s_1 s_2 \dots s_n) = s_1 s_2 \dots s_{i-1} s_j s_{i+1} \dots s_{j-1} s_i s_{j+1} \dots s_n$  と定義する。無向グラフ  $G(V, E)$  に対して  $n$ -トランスポジショングラフ  $T_n = (V, E)$  は  $V = \{(u_1 u_2 \dots u_n) | (u_1 u_2 \dots u_n) \text{ は } 1, 2, \dots, n \text{ の順列} \}$  と  $E = \{(u, TP_{(i,j)}(u)) | u \in V, 1 \leq i < j \leq n\}$  である。

### 7.2 トランスポジショングラフの性質

トランスポジショングラフは再帰性と対称性を持つグラフで表 2 のような性質を持つ。

表 2  $n$ -トランスポジショングラフの性質

頂点数	次数	連結度	直径
$n!$	$n(n-1)/2$	$n(n-1)/2$	$n-1$

## 7.3 トランスポジショングラフの関連研究紹介

### 7.3.1 Node-disjoint paths in a transposition graph

本論文 [11] では  $n$ -トランスポジショングラフの頂点と頂点集合間素な経路選択問題に関する解法を提案している。提案された手法は  $O(n^7)$  の時間計算量を必要とし、最大経路長は  $3n-5$  である。

$n$ -トランスポジショングラフの次数は  $n(n-1)/2$  であり、これは構築が必要な素な経路が  $n(n-1)/2$  個必要であることを意味する。そして、提案手法のキーポイントはグラフの再帰構造を利用している点である。 $n$ -トランスポジショングラフは頂点順列の最右記号  $k(1 \leq k \leq n)$  が同じである順列より誘発される  $n-1$ -サブトランスポジショングラフ  $n$  個に分割することができる。トランスポジショングラフは対称性を持っているため、出発頂点  $s$  を  $12 \dots n$  に固定することができる。これに対して任意の目的頂点の数を  $n(n-1)/2$  とする。経路構築のために出発頂点が存在する  $n-1$  サブトランスポジショングラフの中に目的頂点がいくつ存在するかによって場合を分ける。出発頂点と同一サブグラフの中に  $(n-1)(n-2)/2$  個以上が存在する場合その目的頂点を長さ高々 2 の経路で他のサブグラフへ送り出す。もし、 $(n-1)(n-2)/2$  個以下存在する場合は他のサブグラフに存在する目的頂点を出発頂点が存在するサブグラフへ持ってくる。この操作を全ての経路が構築されるまで繰り返す。

### 7.3.2 Polynomial time algorithm for constructing vertex-disjoint paths in transposition graphs

本論文 [12] では  $n$ -トランスポジショングラフの頂点と頂点集合間素な経路選択問題に関する解法を提案している。該当問題はすでに [10] で解決されたが提案された手法とは違うアプローチを取ることで、問題の解決に要する時間計算量を  $O(n^6)$  に減らし最大経路長も高々最短経路長 +6 に減らしている。

この論文で提案された手法のキーポイントはグラフの頂点に対して Vertical クラスタリング及び Horizontal クラスタリングを行なったことと、クラスタリングされた頂点間を 2 部グラフとみなし、2 部グラフにおける極大マッチングを行っていることである。

## 8 パンケーキグラフ

この節では、パンケーキグラフに関する主な性質および有用な経路選択アルゴリズムに関する研究を紹介する。



## 8.1 パンケーキグラフの定義

$s_1 s_2 \dots s_n$  を 1 から  $n$  までの  $n$  種類の記号で作られる順列とする。 $PR_{(i)}(s_1 s_2 \dots s_n) = s_i s_{i-1} \dots s_1 s_{i+1} s_{i+2} \dots s_n$  と定義する。無向グラフ  $G(V, E)$  に対して  $n$ -パンケーキグラフ  $P_n = (V, E)$  は  $V = \{(u_1 u_2 \dots u_n) | (u_1 u_2 \dots u_n) \text{ は } 1, 2, \dots, n \text{ の順列}\}$  と  $E = \{(u, PR_{(i)}(u)) | u \in V, 2 \leq i \leq n\}$  である。以後  $n$ -パンケーキグラフを  $P_n$  とする。

## 8.2 パンケーキグラフの性質

パンケーキグラフは再帰性と対称性を持つグラフで表 2 のような性質を持つ。

表 3  $n$ -パンケーキグラフの性質

頂点数	次数	連結度	直径
$n!$	$n - 1$	$n - 1$	

## 8.3 パンケーキグラフの関連研究紹介

### 8.3.1 An Algorithm for Node-Disjoint paths in Pancake graphs

本論文 [13] では  $P_n$  における頂点間素な経路選択問題に関する解法を提案している。提案手法の時間計算量は  $O(n^3)$  であり、最大経路長は??である。提案手法ではグラフの対称性利用し出発頂点  $s$  を  $12 \dots n$  に固定する。次に目的頂点  $d$  によって次の 3 つのケースに分ける。

Case 1  $d = d_1 d_2 \dots d_{n-1} n$

Case 2  $d = n d_1 d_2 \dots d_n$

Case 3 *Otherwise.*

素な経路を構築するアルゴリズムのキーポイントはグラフの再帰構造を利用することである。頂点順列の最右記号  $k (1 \leq k \leq n)$  が同じである順列より誘発される  $n - 1$ -部分パンケーキグラフを  $P_{n-1}(k)$  とする。Case 1 の場合は出発頂点と目的頂点が  $P_{n-1}(n)$  に属している状態である。部分グラフ内で構築できる最大の素な経路は  $n - 2$  個であるため、異なる部分グラフを利用し出発頂点と目的頂点を連結するパスを構築後、出発頂点と目的頂点が含まれている部分パンケーキグラフでアルゴリズムを再帰的に適用し  $n - 2$  個の経路を構築する。Case 2 の場合は  $PR_n(d)$  と出発頂点  $s$  と出発頂点が存在する部分パンケーキグラフ内で 1 本の経路を構築後  $n - 2$  個の  $s$  の隣接頂点を  $n - 2$  個の  $P_{n-1}(k) (2 \leq k \leq n)$  他のサブグラフへ移動する経路を構築し、さらにそれぞれのサブグラフより  $P_{n-1}(n)$  に移動する経路を構築する。これにより、 $d$  が属している部分パンケーキグラフで、再度再帰的に提案手法を適用することができる。Case 3 では Case 2 のような  $P_{n-1}(s)$  より  $d$  へ直接いく経路 ( $d$  以外  $P_{n-1}(d)$  の頂点が経路に含まれない) が存在しないため  $P_{n-1}(d_1)$

の頂点を利用し直接いく経路を 1 個  $mn$  構築する。それ以降は Case 2 とほぼ同じく  $n - 2$  個の  $s$  の隣接頂点を他の部分パンケーキグラフを利用し  $d$  が属している部分グラフへの  $n - 2$  個の経路を構築することで  $d$  が属している部分パンケーキグラフで、再度再帰的に提案手法を適用することができる。

### 8.3.2 Node-to-Set Disjoint Paths Problem in Pancake Graphs

本論文 [14] ではパンケーキグラフにおける頂点と頂点集合間素な経路選択問題に関する解法を提案している。提案手法の時間計算量は  $O(n^5)$  であり、最大経路長は??である。提案手法のキーポイントは  $P_n$  の各頂点を  $PR_{n-1}, PR_n$  を交互に適用しすることで得られるクラスに分類したことである。この分類による各クラスは以下の 4 つの重要な特徴を持つ。

1. 各頂点は必ず一つのクラスに属する。
2. 各クラスには  $2n$  個の頂点が存在し、リング構造である。
3. 各  $P_{n-1}$  には同じクラスの頂点がちょうど 2 個存在する。
4. 各頂点は同じクラスに属する二つの隣接頂点を持ち、異なるクラスに属する  $(n - 3)$  個の隣接頂点を持つ。

例えば  $P_4$  の全ての頂点は以下の 3 つのクラスに分類できる。

$$C1 = (1,2,3,4), (4,3,2,1), (2,3,4,1), (1,4,3,2), (3,4,1,2), (2,1,4,3), (4,1,2,3), (3,2,1,4)$$

$$C2 = (2,1,3,4), (4,3,1,2), (1,3,4,2), (2,4,3,1), (3,4,2,1), (1,2,4,3), (4,2,1,3), (3,1,2,4)$$

$$C3 = (1,3,2,4), (4,2,3,1), (3,2,4,1), (1,4,2,3), (2,4,1,3), (3,1,4,2), (4,1,3,2), (2,3,1,4)$$

本論文 [14] の提案手法は二つの手順で構成されている。手順 1 は全ての目的頂点が  $P_{n-1}(n)$  に属している特別な場合を扱っている。この手順では  $n - 2$  個の素な経路提案手法を再帰的に適用することで  $P_{n-1}(n)$  で構築したあと、残り 1 本の経路を  $P_{n-1}(n)$  以外のサブグラフで構築している。手順 2 は手順 1 の場合以外の一般的な場合を扱っている。最初  $P_{n-1}(n)$  に含まれている  $n - 1$  個の頂点より、すべての目的頂点への素な経路を先ほど提案した class の経路を利用し構築する。その後、経路の一部を修正し出発頂点へつなぎ直す。最後に提案手法を再帰的に適用し全体のパスを構築する。

## 9 焦げたパンケーキグラフ

この節では、焦げたパンケーキグラフに関する主な性質および有用な経路選択アルゴリズムに関する研究を紹介する。

## 9.1 焦げたパンケーキグラフ

$s_1 s_2 \dots s_n$  を 1 から  $n$  までの  $n$  種類の記号で作られる符号付き順列とする。符号付き前置反転操作  $SR_{(i)}(s_1 s_2 \dots s_n) = \overline{s_i s_{i-1}} \dots \overline{s_1 s_{i+1} s_{i+2}} \dots s_n$  と定義する。無向グラフ  $G(V, E)$  に対して  $n$ -焦げたパンケーキグラフ  $BP_n = (V, E)$  は  $V = \{(u_1 u_2 \dots u_n) | (u_1 u_2 \dots u_n) \text{ は符号付き } 1, 2, \dots, n \text{ の順列}\}$  と  $E = \{(u, PR_{(i)}(u)) | u \in V, 2 \leq u \leq n\}$  である。以後  $n$ -焦げたパンケーキグラフを  $BP_n$  とする。

## 9.2 焦げたパンケーキグラフの性質

パンケーキグラフは再帰性と対称性を持つグラフで表 2 のような性質を持つ。

表 4  $n$ -焦げたパンケーキグラフの性質

頂点数	次数	連結度	直径
$n! \times 2^n$	$n - 1$	$n - 1$	

## 9.3 焦げたパンケーキグラフの関連研究紹介

### 9.3.1 An Algorithm for Node-to-Set Disjoint Paths Problem in Burnt Pancake Graphs

本論文 [15] では焦げたパンケーキグラフにおける頂点と頂点集合間素な経路選択問題に関する解法を提案している。提案手法の時間計算量は  $O(n^5)$  であり、最大経路長は??である。提案手法のキーポイントは  $BP_n$  の各頂点に対して  $SR_n$  と  $SR_{n-1}$  を交互に適用することで得られる頂点集合を *Traversal class* と定義しその性質を利用していることである。*Traversal class* では次の重要な性質を持つ。

1. 各頂点は必ず一つのクラスに属する。
2. 各クラスには  $4n$  個の頂点が存在し、リング構造である。
3. 各  $BP_{(n-1)}$  には同じクラスの頂点がちょうど 2 個存在する。
4. 各頂点は同じクラスに属する二つの隣接頂点を持ち、残りの隣接頂点は異なるクラスに属する。

たとえば、 $BP_3$  は次の 4 つの *Traversal class* が存在する。

$$\begin{aligned}
 C1 &= \{(1, 2, 3), (\bar{3}, \bar{2}, \bar{1}), (2, 3, \bar{1}), (1, \bar{3}, \bar{2}), (3, \bar{1}, \bar{2}), (2, 1, \bar{3}), \\
 &\quad (\bar{1}, \bar{2}, \bar{3}), (3, 2, 1), (\bar{2}, \bar{3}, 1), (\bar{1}, 3, 2), (\bar{3}, 1, 2), (\bar{2}, \bar{1}, 3)\} \\
 C2 &= \{(\bar{2}, 1, 3), (\bar{3}, \bar{1}, 2), (1, 3, 2), (\bar{2}, \bar{3}, \bar{1}), (3, 2, \bar{1}), (1, \bar{2}, \bar{3}), \\
 &\quad (2, \bar{1}, \bar{3}), (3, 1, \bar{2}), (\bar{1}, \bar{3}, \bar{2}), (2, 3, 1), (\bar{3}, \bar{2}, 1), (\bar{1}, 2, 3)\}
 \end{aligned}$$

$$\begin{aligned}
C3 &= \{(\bar{1}, \bar{2}, 3), (\bar{3}, 2, 1), (\bar{2}, 3, 1), (\bar{1}, \bar{3}, 2), (3, 1, 2), (\bar{2}, \bar{1}, \bar{3}), \\
&\quad (1, 2, \bar{3}), (3, \bar{2}, \bar{1}), (2, \bar{3}, \bar{1}), (1, 3, \bar{2}), (\bar{3}, \bar{1}, \bar{2}), (2, 1, 3)\} \\
C4 &= \{(2, \bar{1}, 3), (\bar{3}, 1, \bar{2}), (\bar{1}, 3, \bar{2}), (2, \bar{3}, 1), (3, \bar{2}, 1), (\bar{1}, 2, \bar{3}), \\
&\quad (\bar{2}, 1, \bar{3}), (3, \bar{1}, 2), (1, \bar{3}, 2), (\bar{2}, 3, \bar{1}), (\bar{3}, 2, \bar{1}), (1, \bar{2}, 3)\}
\end{aligned}$$

$BP_n$  の頂点と頂点集合間の素な経路選択アルゴリズムは 1. すべての目的頂点が出発頂点と同じ部分グラフ  $BP_{n-1}(n)$  に属している場合と、2.1 以外の場合に分けて解法を提案している。1 の場合では  $BP_{n-1}(n)$  で提案手法を再帰的に適用し、 $n - 1$  個の素な経路を構築する。そして、残りの 1 本の経路を  $BP_{n-1}(n)$  以外の部分グラフを用いて経路を構築する。2 の場合ではすべての目的頂点グループを  $BP_{n-1}(n)$  *Traversal class* 上に存在し、かつ他の目的頂点を含まない場合  $D1$  とそうでない場合に  $D2$  わける。分けられたグループに対して、 $D1$  はそのまま出発頂点への経路を構築し  $D2$  に対しては、*Traversal class* 上で目的頂点の経路上での重複が避けられるように経路を再構築しながら出発頂点への経路を構築する。

10 まとめ

11 おわりに

## 参考文献

- [1] Top 500, <https://www.top500.org/lists/2016/11/>, Jan. 2017.
- [2] Mitsuo Yokokawa, et al. "The K computer: Japanese next-generation supercomputer development project," *Low Power Electronics and Design (ISLPED) 2011 International Symposium on. IEEE*, 2011.
- [3] Oakforest-PACS スーパーコンピュータシステム, <http://www.cc.u-tokyo.ac.jp/system/ofp/>, Jan. 2017.
- [4] Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you?, [https://www.usenix.org/legacy/events/fast07/tech/schroeder/schroeder\\_html/index.html](https://www.usenix.org/legacy/events/fast07/tech/schroeder/schroeder_html/index.html), Jan.2017.
- [5] Sanguthevar Rajasekaran and David S. L. Wei, "Selection, Routing, and Sorting on the Star Graph," *Journal of Parallel and Distributed Computing*, vol.42,no.2,pp225-233, 1997.
- [6] S.G. Akl and K. Qiu, "Data Communication and Computational Geometry on the Star and Pancake Interconnection Networks," *Parallel and Distributed Processing, 1991. Proceedings of the Third IEEE Symposium*, pp415-422,1991.
- [7] Qian-Ping Gu and Shietung Peng, "Node-to-node cluster fault tolerant routing in star graphs," *Information Processing Letters*, vol.56,pp29-35, 1995.
- [8] M. Dietzfelbinger, S. Madhavapeddy and I.H. Sudborough, "Three disjoint path paradigms in star networks," *Proc. IEEE Symp. on Parallel and Distributed Processing*, pp400-406, 1991.
- [9] Qian-Ping Gu and Shietung Peng, "Node-to-set disjoint paths problem in star graphs," *Information Processing Letters*, vol.62,pp201-207, 1996.
- [10] S. Latifi and P. K. Srimani, "Transposition networks as a class of fault-tolerant robust networks," *IEEE Transactions on Computers*, vol.45, no.2, pp230-238, 1996.
- [11] Y. Suzuki, K. Kaneko and M. Nakamori, "Node-disjoint paths in a transposition graph," *Proceedings of the International Conference on Parallel and Distributed, Processing Techniques and Applications*, pp. 298-304, 2004.
- [12] Satoshi Fujita, "Polynomial time algorithm for constructing vertex-disjoint paths in transposition graphs," *Networks*, Volume 56, pp. 149-157, September 2010.
- [13] , Keiichi Kaneko, and Yasuto Suzuki, "An Algorithm for Node-Disjoint Paths in Pancake Graphs," *IEICE Trans. Information and Systems*, Vol. E86-D, No. 3, pp. 610-615, Mar. 2003
- [14] Keiichi Kaneko, and Yasuto Suzuki, "Node-to-Set Disjoint Paths Problem in Pancake Graphs," *IEICE Trans. Information and Systems*, Vol. E86-D, No. 9, pp. 1628-1633, Sept. 2003.

- [15] Keiichi 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