

# 特別研究計画

鄭 信雨

学籍番号 11834304

2017 年 1 月 22 日

## 目次

1	はじめに	4
2	目的	4
3	背景	4
4	諸定義	5
5	様々なケイリーグラフ	8
5.1	ハイパーキューブ (hypercube) $Q_n$	8
5.2	スターグラフ (star graph) $S_n$	9
5.3	パンケーキグラフ (pancake graph) $P_n$	10
5.4	焦げたパンケーキグラフ (burnt pancake graph) $BP_n$	11
5.5	ローテータグラフ (rotator graph) $R_n$	12
5.6	バイローテータグラフ (rotator graph) $BR_n$	13
5.7	トランスポジショングラフ (transposition graph) $T_n$	14
5.8	部分文字列反転グラフ (substring reversal graph) $SR_n$	15
5.9	バブルソートグラフ (bubble sort graph) $B_n$	16
6	グラフの性能および素な経路選択解法の有効性	17
6.1	集積度	17
6.2	素な経路選択アルゴリズムの有効性	17
7	スターグラフ	19
7.1	スターグラフ関連研究紹介	19
8	バブルソートグラフ	21
9	トランスポジショングラフ	21
9.1	トランスポジショングラフの定義	21
9.2	トランスポジショングラフの性質	21
9.3	トランスポジショングラフの関連研究紹介	21
10	パンケーキグラフ	22
10.1	パンケーキグラフの関連研究紹介	22
11	焦げたパンケーキグラフ	24

11.1	焦げたパンケーキグラフ . . . . .	24
11.2	焦げたパンケーキグラフの関連研究紹介 . . . . .	24
12	まとめ	26
13	おわりに	26

## 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。

## 4 諸定義

本章では、本レポートで使用する用語の定義を行う。

**定義 1.** グラフ (graph) は頂点 (vertex, node) を要素と持つ空でない集合と、二つの頂点を要素とする辺 (edge, link) を要素と持つ集合からなる。頂点集合が  $V$ , 辺集合が  $E$  であるグラフ  $G$  を  $G(V, E)$  と書く。

**定義 2.** 各辺を定義する頂点の対が順序対であるとき、そのグラフを有向グラフ (directed graph), 非順序対のとき、無向グラフ (undirected graph) と呼ぶ。

**定義 3.** 二つの頂点  $u, v$  と辺  $e$  に対して、 $u, v \in e$  のとき、 $e$  は  $u, v$  に接続しているという。また、 $u, v$  は隣接している (adjacent),  $e$  の端点 (end point) は  $u, v$  である、という。以後、頂点  $v$  の隣接頂点集合を  $N(v)$  で表す。

**定義 4.** 無向グラフの頂点  $v$  において、 $v$  に接続している辺の数を  $v$  の次数 (degree) という。次数が 0 である頂点を孤立頂点 (isolated vertex) という。

**定義 5.** 無向グラフ  $G$  において、任意の頂点の次数のうち、最大のものを  $G$  の最大次数 (maximum degree), 最小のものを  $G$  の最小次数 (minimum degree) という。

**定義 6.** 有向グラフの頂点  $v$  において、 $v$  から出る辺の集合を  $out(v)$ ,  $v$  へ入る辺の集合を  $in(v)$  とする。  $|out(v)|$ ,  $|in(v)|$  をそれぞれ  $v$  の出次数 (out degree), 入次数 (in degree) という。出次数, 入次数が共に 0 である頂点を孤立頂点という。

**定義 7.** 全ての頂点の次数が等しいグラフを正則グラフ (regular graph) という。

**定義 8.** グラフ  $G$  の任意の二頂点  $u, v$  に対し、 $u$  を  $v$  へ写す  $G$  の自己同型写像があるとき、 $G$  を頂点对称 (vertex transitive, vertex symmetric) という。 $G$  の任意の二辺  $e_1, e_2$  に対し、 $e_1$  を  $e_2$  へ写す  $G$  への自己同型写像があるとき、 $G$  を辺対称 (edge transitive, edge symmetric) という。

**定義 9.** 無向グラフ  $G(V, E)$  の  $k$  個の頂点の系列  $v_1, v_2, \dots, v_k$  が  $1 \leq i \leq k-1$  となる  $i$  に対して  $(v_i, v_{i+1}) \in E$  を満たすとき、この系列を  $G$  の  $v_1$  から  $v_k$  への長さ  $k-1$  の経路 (path) という。有向グラフにおいては、有向経路 (directed path) という。

**定義 10.** 先頭の頂点と最後の頂点が同じ経路を閉路 (cycle) という。

**定義 11.** 両端の頂点が同じ辺を自己閉路 (self loop) という。

**定義 12.** 有向グラフ  $G$  において、各辺の向きを無くして得られるグラフを  $G$  の基礎グラフ (underlying graph) という。

**定義 13.** 任意の二頂点間に有向経路が存在する有向グラフを強連結 (strongly connected) という. 任意の二頂点間に経路が存在する無向グラフを連結 (connected) という. 有向グラフは, その基礎グラフが連結であるとき, 連結という. 強連結でも連結でもないグラフを非連結 (unconnected) という.

**定義 14.** 閉路を含まない連結なグラフを木 (tree) という.

**定義 15.** 無向グラフにおいて, 隣接するある二頂点間に複数の辺が存在するとき, これらの辺を多重辺 (multiple edge) という. 有向グラフにおいて, 隣接するある二頂点  $u, v$  間に,  $u$  から  $v$  への複数の辺が存在する場合, これらの辺を多重辺という.

**定義 16.** 自己閉路と多重辺を持たないグラフを単純グラフ (simple graph) という.

**定義 17.** グラフ  $G$  の二頂点  $u, v$  間の経路のうち, 長さが最も短いものを  $G$  における  $u, v$  間の最短経路 (shortest path) という.

**定義 18.** 二頂点  $u, v$  間の最短経路の長さを  $u, v$  間の距離 (distance) という.  $u, v$  間に経路が存在しない場合は,  $u, v$  間の距離を  $\infty$  とする.

**定義 19.** 連結グラフ  $G$  の任意の二頂点間の距離のうち, 最大のものを  $G$  の直径 (diameter) という. 以降グラフ  $G$  の直径を  $D(G)$  で表す.

**定義 20.** グラフ  $G(V, E)$  に対し, 次の条件を全て満たすグラフ  $G'(V', E')$  を  $G$  の部分グラフ (subgraph) という.

- $V' \subseteq V$
- $E' \subseteq E$

**定義 21.** 部分グラフのうち, 木であるものを部分木 (subtree) という.

**定義 22.** グラフ  $G$  の部分木のうち, その頂点が  $G$  のすべての頂点からなるものを  $G$  の全域木 (spanning tree) という.

**定義 23.** 連結なグラフ  $G$  に対して, 以下の二つの条件をともに満たす最小の  $k$  を  $G$  の連結度 (connectivity) という.

- $G$  の任意の  $k - 1$  個の頂点を取り除いたグラフは連結である.
- $G$  のある  $k$  個の頂点を取り除いたグラフが非連結に, あるいはたったひとつの頂点からなるグラフになる.

**定義 24.** 連結度が  $k$  であるグラフを  $k$ -連結 ( $k$ -connected) という.

**定義 25.** 複数の経路が頂点を共有しないとき, これらの経路を互いに素な経路 (disjoint path) という.

**定義 26.** 二頂点  $u, v$  を結ぶ複数の経路が  $u, v$  を除いて頂点を共有しないとき, これらの経路を内素な経路 (internally-disjoint paths) もしくは頂点間素な経路 (node-to-node disjoint paths) という.

**定義 27.** 頂点  $u$  と  $u$  を含まない頂点集合  $V$  を結ぶ複数の経路が  $u$  を端点とした場合を除いて頂点を共有しないとき, これらの経路を頂点と頂点集合間素な経路 (node-to-set disjoint paths) という.

**定義 28.** 頂点  $v_1, \dots, v_n$  を要素として持つ集合  $V$  と  $V$  の頂点を含まない頂点  $u_1, \dots, u_n$  を要素として持つ集合  $U$  に対して  $u_i, v_j$  ( $1 \leq i, j \leq n$ ) 間を結ぶ複数の経路が頂点を共有しないとき, これらの経路を頂点集合間素な経路 (set-to-set disjoint paths) という.

**定義 29.** 集合  $B$  に演算  $\circ$  が定義されていて次の性質を全て満たす時,  $\langle B, \circ \rangle$  を群 (group) という.

- $B$  の任意の元  $x, y, z$  に対して,

$$(x \circ y) \circ z = x \circ (y \circ z)$$

を満たす.

- $B$  の任意の元  $x$  に対して,

$$x \circ e = e \circ x = x$$

を満たす  $B$  の元  $e$  が存在する. この  $e$  を群  $\langle B, \circ \rangle$  の単位元 (unit element, identity) という.

- $B$  の任意の元  $x, y, z$  に対して,

$$x \circ x^{-1} = x^{-1} \circ x = e$$

を満たす  $B$  の元  $x^{-1}$  が存在する.

**定義 30.**  $B = \langle B, \circ \rangle$  を群とし,  $B$  の元の部分集合を  $X$  とする.  $B$  の任意の元を  $X$  の元と演算  $\circ$  の組み合わせで表すことができるとき,  $X$  を  $B$  の生成元集合 (generator set) という.

**定義 31.**  $B = \langle B, \circ \rangle$  を群,  $X$  を  $B$  の生成元集合とする. 有向グラフ  $C * (B, X)$  は, その頂点集合  $V(C * (B, X))$  と辺集合  $E(C * (B, X))$  をそれぞれ

$$\begin{aligned} V(C * (B, X)) &= B \\ E(C * (B, X)) &= \{\{b, x \circ b\} | b \in B, x \in X\} \end{aligned}$$

とするとき, 有向ケイリーグラフと呼ばれる.

**定義 32.**  $B = \langle B, \circ \rangle$  を群,  $X$  を  $B$  の生成元集合とする.  $X$  の任意の要素  $x$  に対して  $x^{-1} \in X$  であるとする. 無向グラフ  $C(B, X)$  はその頂点集合  $V(C * (B, X))$  と辺集合  $E(C * (B, X))$  をそれぞれ

$$\begin{aligned} V(C * (B, X)) &= B \\ E(C * (B, X)) &= \{\{b, x \circ b\} | b \in B, x \in X\} \end{aligned}$$

とすると、ケイリーグラフと呼ばれる。



## 5 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.

### 5.1 ハイパーキューブ (hypercube) $Q_n$

$\mathbf{u} = (u_1, u_2, \dots, u_n)$  を 0 と 1 からなる作られるビット列とする. 整数  $i (1 \leq i \leq n)$  に対して記号反転操作  $Q_i(\mathbf{u})$  を次のように定義する.

$$Q_i(\mathbf{u}) = (u_1, u_2, \dots, (u_i + 1) \bmod 2, \dots, u_n)$$

無向グラフ  $G(V, E)$  に対して  $n$ -hypercube  $Q_n = (V, E)$  の  $V, E$  を以下に示す.

$$V = \{u_1 u_2 \dots u_n \mid 0 \text{ と } 1 \text{ からなる作られる長さ } n \text{ の全てのビット列} \}$$

$$E = \{(\mathbf{u}, Q_i(\mathbf{u})) \mid \mathbf{u} \in V, 1 \leq i \leq n\}$$

以後  $n$ -ハイパーキューブ を  $Q_n$  とする. 図 1 に  $n$  が 4 の場合のハイパーキューブ を示す. 表 1 に  $Q_n$  の性質を示す.

表 1  $n$ -ハイパーキューブ の性質

単純グラフ	再帰性	対称性	頂点数	次数	連結度	直径
yes	yes	yes	$2^n$	$n$	$n$	$n$

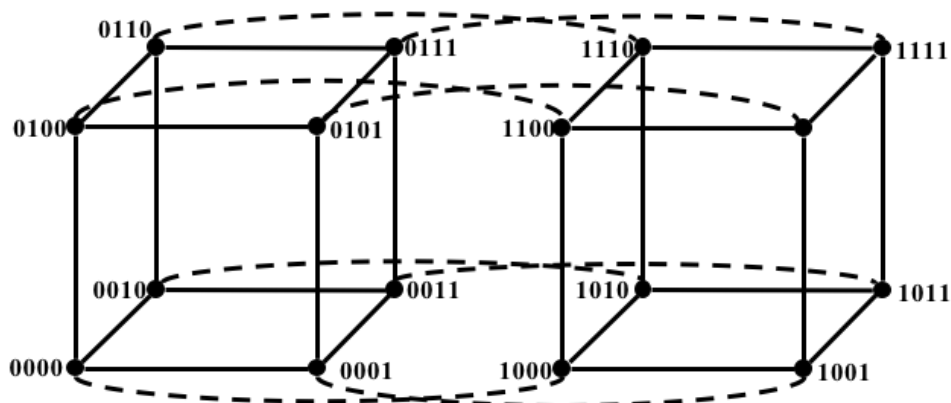


図 1 4-ハイパーキューブ

## 5.2 スターグラフ (star graph) $S_n$

$\mathbf{u} = (u_1, u_2, \dots, u_n)$  を 1 から  $n$  までの  $n$  種類の記号で作られる順列とする.  $S_j(u_1 u_2 \dots u_n) = u_j, u_2 \dots u_{j-1} u_1 u_{j+1} \dots u_n$  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.

$$V = \{u_1 u_2 \dots u_n \mid 1, 2, \dots, n \text{ からなる全ての順列}\}$$

$$E = \{(\mathbf{u}, S_i(\mathbf{u})) \mid \mathbf{u} \in V, 2 \leq i \leq n\}$$

以後  $n$ -スターグラフを  $S_n$  とする. 図 2 に  $n$  が 4 の場合のスターグラフを示す. 表 2 にスターグラフの性質を示す.

表 2  $n$ -スターグラフの性質

単純グラフ	再帰性	対称性	頂点数	次数	連結度	直径
yes	yes	yes	$n!$	$n - 1$	$n - 1$	$\lfloor 3(n - 1)/2 \rfloor$

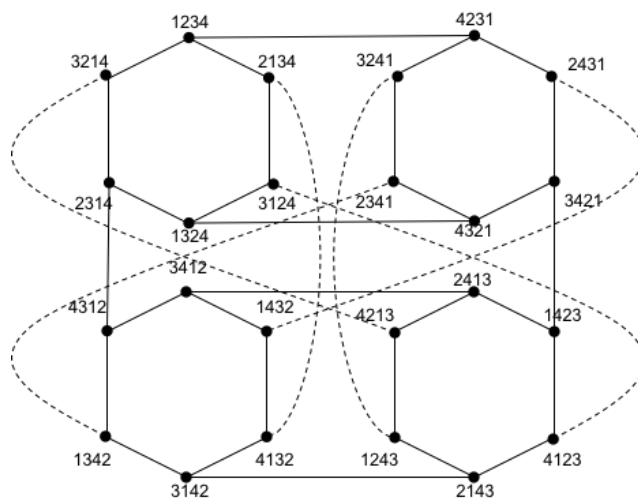


図 2 4-スターグラフ

### 5.3 パンケーキグラフ (pancake graph) $P_n$

$\mathbf{u} = (u_1, u_2, \dots, u_n)$  を 1 から  $n$  までの  $n$  種類の記号で作られる順列とする.  $PR_i(u_1 u_2 \dots u_n) = u_i u_{i-1} \dots u_1 u_{i+1} u_{i+2} \dots u_n$  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  $P_n = (V, E)$  の  $V, E$  を以下に示す.

$$V = \{u_1 u_2 \dots u_n \mid 1, 2, \dots, n \text{ からなる全ての順列}\}$$

$$E = \{(\mathbf{u}, PR_i(\mathbf{u})) \mid \mathbf{u} \in V, 2 \leq i \leq n\}$$

以後  $n$ -パンケーキグラフを  $P_n$  とする. 図 3 に  $n$  が 4 の場合のパンケーキグラフを示す. 表 3 に  $P_n$  の性質を示す.

表 3  $n$ -パンケーキグラフの性質

単純グラフ	再帰性	対称性	頂点数	次数	連結度	直径
yes	yes	yes	$n!$	$n - 1$	$n - 1$	$\leq \lceil 5(n + 1)/3 \rceil$

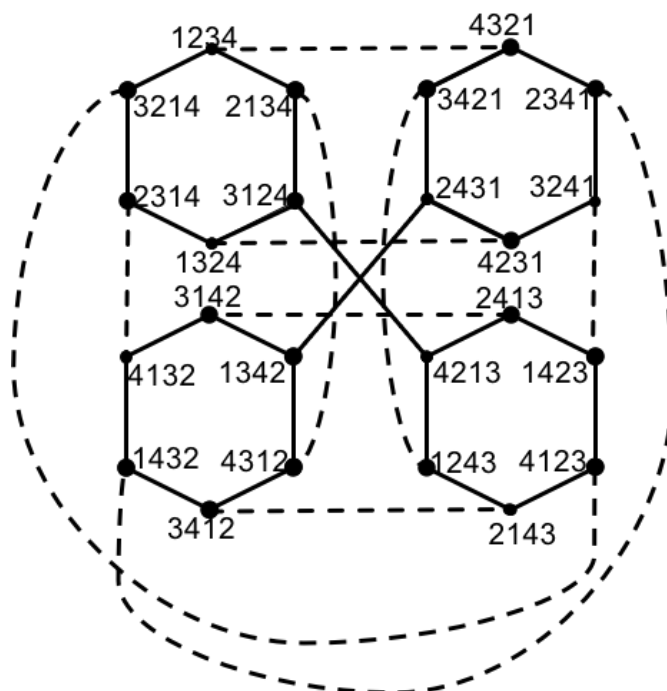


図 3 4 - パンケーキグラフ

## 5.4 焦げたパンケーキグラフ (burnt pancake graph) $BP_n$

$u = (u_1, u_2, \dots, u_n)$  を 1 から  $n$  までの  $n$  種類の符号付き記号で作られる順列とする. 次に符号付き前置反転操作  $SR_i(u_1 u_2 \dots u_n) = \overline{u_i} u_{i-1} \dots \overline{u_1} u_{i+1} u_{i+2} \dots u_n$  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  $BP_n = (V, E)$  の  $V, E$  を以下に示す.

$$V = \{u_1 u_2 \dots u_n \mid 1, 2, \dots, n \text{ からなる全ての順列}\}$$

$$E = \{(u, PR_i(u)) \mid u \in V, 1 \leq i \leq n\}$$

以後  $n$ -焦げたパンケーキグラフを  $BP_n$  とする. 図 4 に  $n$  が 3 の場合の焦げたパンケーキグラフを示す. 表 4 に  $BP_n$  の性質を示す.

表 4  $n$ -焦げたパンケーキグラフの性質

単純グラフ	再帰性	対称性	頂点数	次数	連結度	直径
yes	yes	yes	$n! \times 2^n$	$n$	$n$	$\leq 2n + 3$

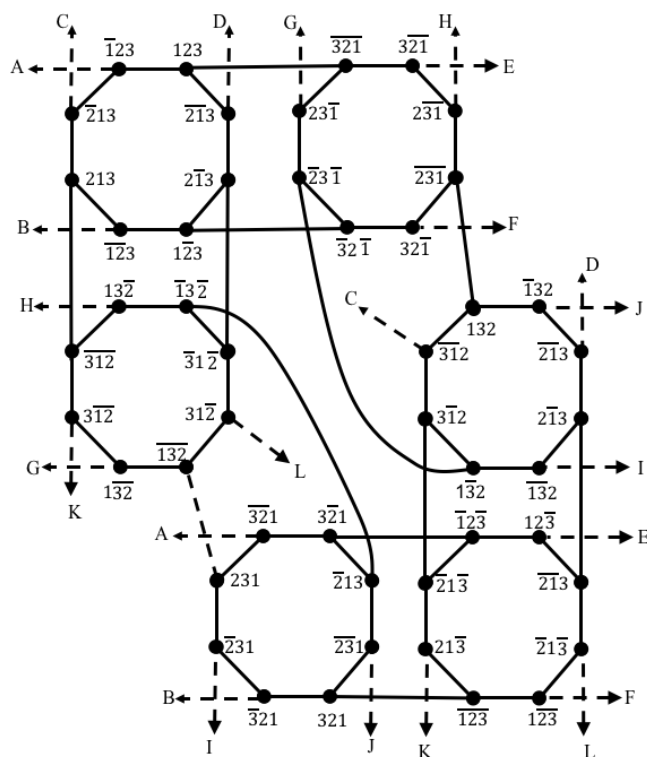


図 4 3 - 焦げたパンケーキグラフ

## 5.5 ローテータグラフ (rotator graph) $R_n$

$\mathbf{u} = (u_1, u_2, \dots, u_n)$  を 1 から  $n$  までの  $n$  種類の記号で作られる順列とする. 有向グラフ  $G(V, E)$  に対して  $n$ -ローテータグラフ  $R_n = (V, E)$  の  $V, E$  を以下に示す。

$$V = \{u_1 u_2 \dots u_n \mid 1, 2, \dots, n \text{ からなる全ての順列} \}$$

$$E = \{(\mathbf{u}, \mathbf{v}) \mid \mathbf{u} \in V, \mathbf{v} \in V, \mathbf{u} = (u_1, u_2, \dots, u_n), \mathbf{v} = (v_1, v_2, \dots, v_n) \text{ の場合} \\ \mathbf{u} \text{ に対して } \mathbf{u} \text{ から } \mathbf{v} \text{ への有向辺が}$$

$$v_1 = u_2, v_2 = u_3, \dots, v_{i-1} = u_i, v_i = u_1, v_{i+1} = u_{i+1}, \\ \dots, v_n = u_n \text{ を満たす } i (2 \leq i \leq n) \text{ があれば存在} \}$$

以後  $n$ -ローテータグラフを  $R_n$  とする。図 5 に  $n$  が 3 の場合のローテータグラフを示す。表 5 に  $R_n$  の性質を示す。

表 5  $n$ -ローテータグラフの性質

単純グラフ	再帰性	対称性	頂点数	次数	連結度	直径
yes	yes	yes	$n!$	$n - 1$	$n - 1$	$n - 1$

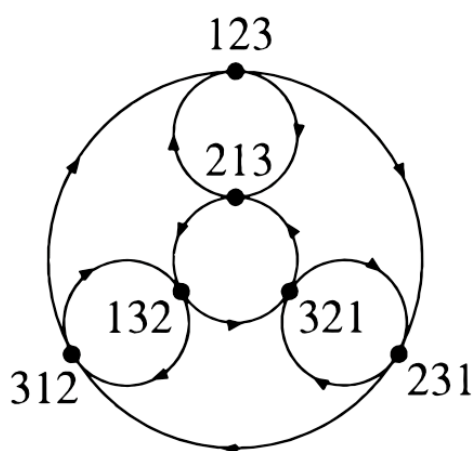


図 5 3 - ローテータグラフ

## 5.6 バイローテータグラフ (rotator graph) $BR_n$

$\mathbf{u} = (u_1 u_2 \dots u_n)$  を 1 から  $n$  までの  $n$  種類の記号で作られる順列とする. 整数  $i (2 \leq i \leq n)$  に対して正のローテーション操作  $R_i^+(\mathbf{u})$  と負のローテーション操作  $R_i^-(\mathbf{u})$  を次のように定義する.

$$R_i^+(\mathbf{u}) = (u_2, u_3, \dots, u_i, u_1, u_{i+1}, u_{i+2}, \dots, u_n)$$

$$R_i^-(\mathbf{u}) = (u_i, u_1, u_2, \dots, u_{i-1}, u_{i+1}, u_{i+2}, \dots, u_n)$$

有向グラフ  $G(V, E)$  に対して  $n$ -バイローテータグラフ  $BR_n = (V, E)$  の  $V, E$  を以下に示す.

$$V = \{u_1 u_2 \dots u_n \mid 1, 2, \dots, n \text{ からなる全ての順列}\}$$

$$E = \{(\mathbf{u}, \mathbf{v}) \mid \mathbf{u} \in V, \mathbf{v} \in V, \mathbf{v} = R_i^+(\mathbf{u}) \text{ or } \mathbf{v} = R_i^-(\mathbf{u}), 2 \leq i \leq n\}$$

以後  $n$ -バイローテータグラフを  $BR_n$  とする. 図 6 に  $n$  が 3 の場合のローテータグラフを示す. 表 6 に  $BR_n$  の性質を示す.

表 6  $n$ -バイローテータグラフの性質

単純グラフ	再帰性	対称性	頂点数	次数	連結度	直径
yes	yes	yes	$n!$	$2n - 3$	$2n - 3$	$n - 1$

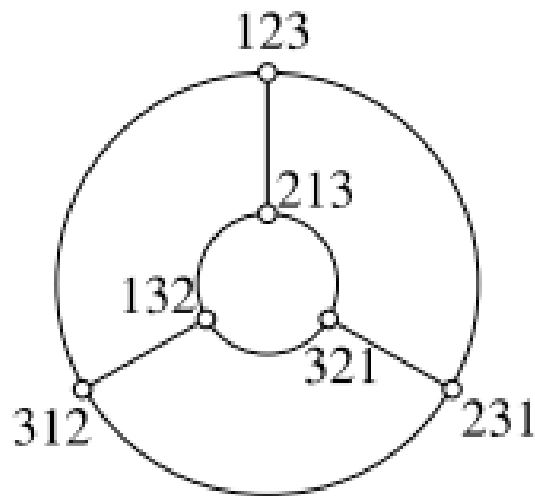


図 6 3 - バイローテータグラフ

## 5.7 トランスポジショングラフ (transposition graph) $T_n$

$\mathbf{u} = (u_1, u_2, \dots, u_n)$  を 1 から  $n$  までの  $n$  種類の記号で作られる順列とする. 整数  $i, j (1 \leq i < j \leq n)$  に対してトランスポジション操作  $T_{i,j}(\mathbf{u})$  を次のように定義する.

$$T_{i,j}(\mathbf{u}) = (u_1, u_2, \dots, u_{i-1}, u_j, u_{i+1}, \dots, u_{j-1}, u_i, u_{j+1}, \dots, u_n)$$

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  $T_n = (V, E)$  の  $V, E$  を以下に示す.

$$V = \{u_1 u_2 \dots u_n \mid 1, 2, \dots, n \text{ からなる全ての順列}\}$$

$$E = \{(\mathbf{u}, T_{i,j}(\mathbf{u})) \mid \mathbf{u} \in V, 1 \leq i < j \leq n\}$$

以後  $n$ -トランスポジショングラフを  $T_n$  とする. 図 7 に  $n$  が 4 の場合のトランスポジショングラフを示す. 表 11 に  $T_n$  の性質を示す.

表 7  $n$ -トランスポジショングラフの性質

単純グラフ	再帰性	対称性	頂点数	次数	連結度	直径
yes	yes	yes	$n!$	$n(n-1)/2$	$n(n-1)/2$	$n-1$

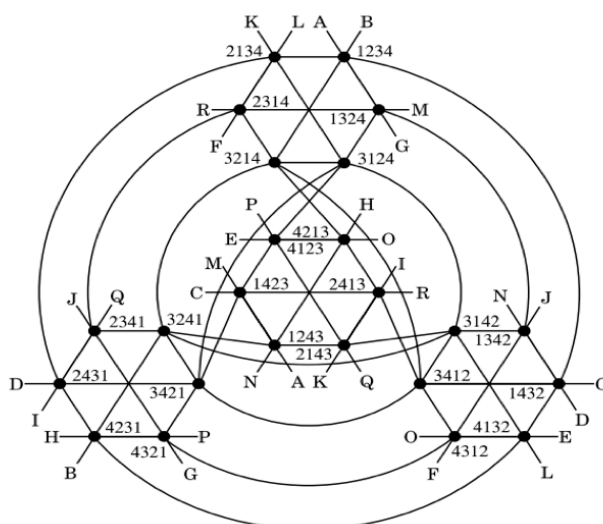


図 7 4 - トランスポジショングラフ

## 5.8 部分文字列反転グラフ (substring reversal graph) $SR_n$

$\mathbf{u} = (u_1 u_2 \dots u_n)$  を 1 から  $n$  までの  $n$  種類の記号で作られる順列とする. 整数  $i, j (1 \leq i < j \leq n)$  に対して部分文字列反転操作  $SR_{i,j}(\mathbf{u})$  を次のように定義する.

$$SR_{i,j}(\mathbf{u}) = (u_1, u_2, \dots, u_{i-1}, u_j, u_{j-1}, \dots, u_{i+1}, u_i, u_{j+1}, \dots, u_n)$$

無向グラフ  $G(V, E)$  に対して  $n$ -部分文字列反転グラフ  $SR_n = (V, E)$  の  $V, E$  を以下に示す.

$$V = \{u_1 u_2 \dots u_n \mid 1, 2, \dots, n \text{ からなる全ての順列}\}$$

$$E = \{(\mathbf{u}, SR_{i,j}(\mathbf{u})) \mid \mathbf{u} \in V, 1 \leq i < j \leq n\}$$

以後  $n$ -部分文字列反転グラフを  $SR_n$  とする. 図 8 に  $n$  が 4 の場合の部分文字列反転グラフを示す. 表 8 に  $SR_n$  の性質を示す.

表 8  $n$ -部分文字列反転グラフの性質

単純グラフ	再帰性	対称性	頂点数	次数	連結度	直径
yes	yes	yes	$n!$	$(n-1)(n-2)/2$	$(n-1)(n-2)/2$	$\leq n-1$

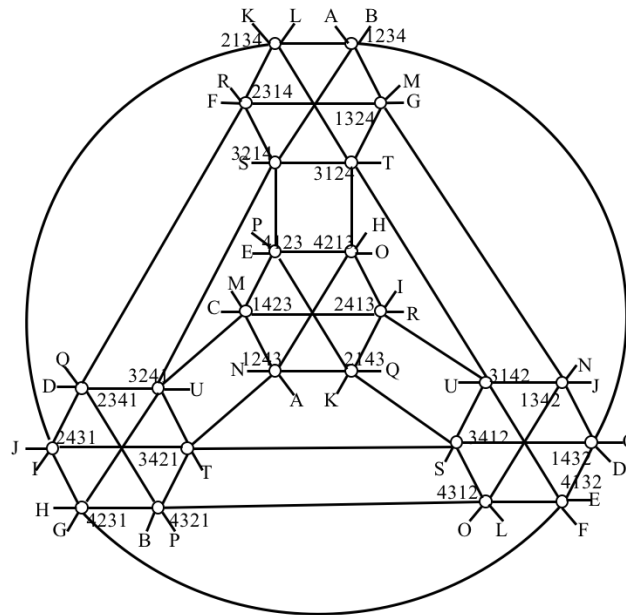


図 8 4 - 部分文字列反転グラフ



## 5.9 バブルソートグラフ (bubble sort graph) $B_n$

$\mathbf{u} = (u_1, u_2, \dots, u_n)$  を 1 から  $n$  までの  $n$  種類の記号で作られる順列とする. 整数  $i (1 \leq i \leq n-1)$  に対して隣接交換操作  $B_i(\mathbf{u})$  を次のように定義する.

$$B_i(\mathbf{u}) = (u_1, u_2, \dots, u_{i-1}, u_{i+1}, u_i, u_{i+1}, \dots, u_n)$$

無向グラフ  $G(V, E)$  に対して  $n$ -バブルソートグラフ  $B_n = (V, E)$  の  $V, E$  を以下に示す.

$$V = \{u_1 u_2 \dots u_n \mid 1, 2, \dots, n \text{ からなる全ての順列}\}$$

$$E = \{(\mathbf{u}, B_i(\mathbf{u})) \mid \mathbf{u} \in V, 1 \leq i \leq n-1\}$$

以後  $n$ -バブルソートグラフを  $B_n$  とする. 図 9 に  $n$  が 4 の場合のバブルソートグラフを示す. 表 9 に  $B_n$  の性質を示す.

表 9  $n$ -バブルソートグラフの性質

単純グラフ	再帰性	対称性	頂点数	次数	連結度	直径
yes	yes	yes	$n!$	$n-1$	$n-1$	$(n-1)n/2$

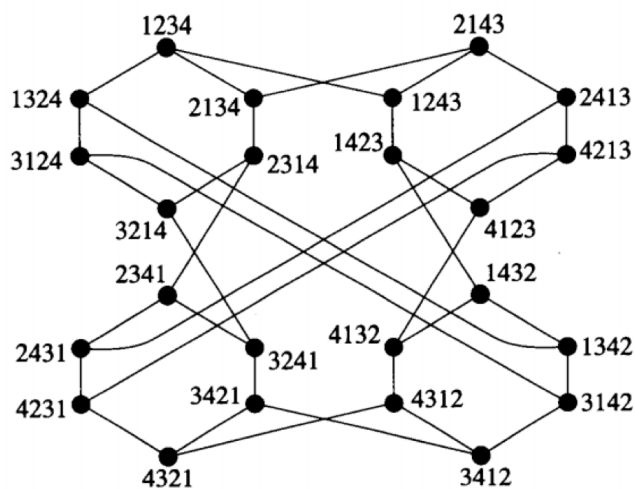


図 9 4-バブルソートグラフ

## 6 グラフの性能および素な経路選択解法の有効性

### 6.1 集積度

並列分散計算システムでは個々のノードの性能も重要であるが、システムの位相を与える相互結合網の性能もとても重要である。その結果相互結合網に関する研究が活発に行われるようになり、先ほど5節で紹介したようなさまざまな総結合結合網が提案されてきた。相互結合網の性能を表す集積度はしばしば以下の式で表される。

$$\text{集積度} = \frac{\text{頂点数}}{(\text{次数}) \times (\text{直径})}$$

相互結合網における次数はノードに繋がる通信路を意味するのでノードに対する通信路の増加はハードウェアの実装に大きく影響する。また、直径は任意のノード間の通信コストと言えることから直径が長くなればなるほど通信コストが増えるため、なるべく小さい直径が望ましい。表10に様々なケーイーリグラフにおける集積度を示す。

表10 様々なケーイーリグラフの集積度

グラフ	頂点数	次数	直径	集積度
$Q_n$	$2^n$	$n$	$n$	$\frac{2^n}{n^2}$
$S_n$	$n!$	$n-1$	$\lfloor 3(n-1)/2 \rfloor$	$\frac{n!}{(n-1)(\lfloor 3(n-1)/2 \rfloor)}$
$P_n$	$n!$	$n-1$	$\leq \lceil 5(n+1)/3 \rceil$	$\frac{n!}{(n-1)(\lceil 5(n+1)/3 \rceil)}$
$BP_n$	$n! \times 2^n$	$n$	$\leq 2n+3$	$\frac{n! \times 2^n}{n(2n+3)}$
$R_n$	$n!$	$n-1$	$n-1$	$\frac{n!}{(n-1)^2}$
$BR_n$	$n!$	$2n-3$	$n-1$	$\frac{n!}{(2n-3)(n-1)}$
$T_n$	$n!$	$\frac{n(n-1)}{2}$	$n-1$	$\frac{2n!}{(n-1)^2 \times n}$
$S_n$	$n!$	$\frac{n(n-1)}{2}$	$\leq n-1$	$\frac{2n!}{(n-1)^2 \times n}$
$B_n$	$n!$	$n-1$	$\frac{n(n-1)}{2}$	$\frac{2n!}{(n-1)^2 \times n}$

### 6.2 素な経路選択アルゴリズムの有効性

先ほど述べたように現在様々な位相の相互結合網が提案されている。これらの相互結合網におけるノード間の通信速度向上や耐故障性の向上のためにグラフ理論の様々な問題を適用し解決してきた。例えば最短経路問題、ハミルトン閉路問題、頂点彩色問題、最小フィードバック頂点集合問題問題などがある。そして、今回このレポートでは頂点間、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主な研究をまとめて述べることを目的としている。

並列分散計算システムでは多数のノードを扱うため一部ノードの故障が頻繁に発生することを予

想できる．このように一部のノードが故障した状態でもシステムは可能な限り正常に動作する必要がある．図 10 に示すように故障が発生した場合でも故障ノードを迂回し通信経路を確保する必要がある．これを耐故障性という．

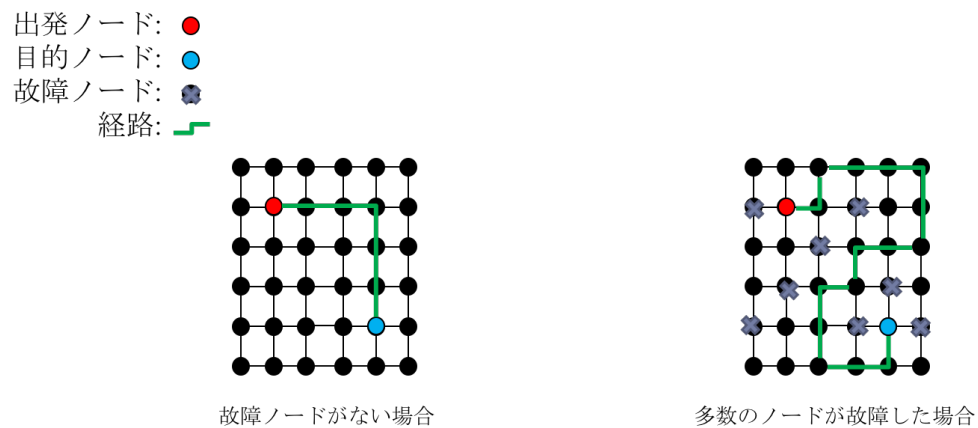


図 10 耐故障経路選択

また，ノード間通信を行う際にはなるべく短時間で通信を行うことが望ましいのは自明である．しかし，単一通信経路の速度向上は物理的な限界があるため可能な限り並列な複数経路を確保することで通信速度向上を達成することができる．図 11 にノード間で並列通信経路が確保された様子を示す．

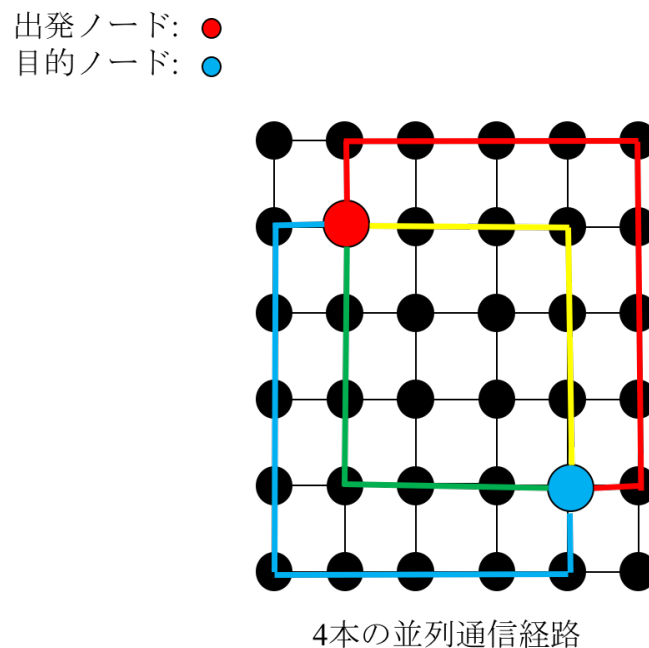


図 11 並列通信経路確保

上記図で示したとおり、各ノードを頂点にノード間のリンクを辺で表現することでグラフ理論として問題を扱うことができる。そして、上記二つの問題は素な経路選択問題を解くことで解決することが可能になる。素な経路選択問題は下記のような分類をすることができる。

- 頂点間素な経路選択
- 頂点と頂点集合間素な経路選択
- 頂点集合間素な経路選択
- 頂点ペア間素な経路選択

まず、素な経路とは複数の経路が同じ頂点を共有しないことである。そして頂点間素な経路選択は任意の2頂点間で複数の経路が経路の両端点を除き互いに素である場合、頂点と頂点集合間素な経路選択は任意の出発頂点と目的頂点集合間の複数の経路が出発頂点を除き互いに素である場合、頂点集合間素な経路選択は任意の出発頂点集合と目的頂点の集合間の複数の経路が互いに素である場合である。最後に頂点ペア間素な経路選択とは出発頂点と目的頂点のペア複数与えられた場合それらペア間の経路が互いに素である場合である。図12にこれらの例を示す。

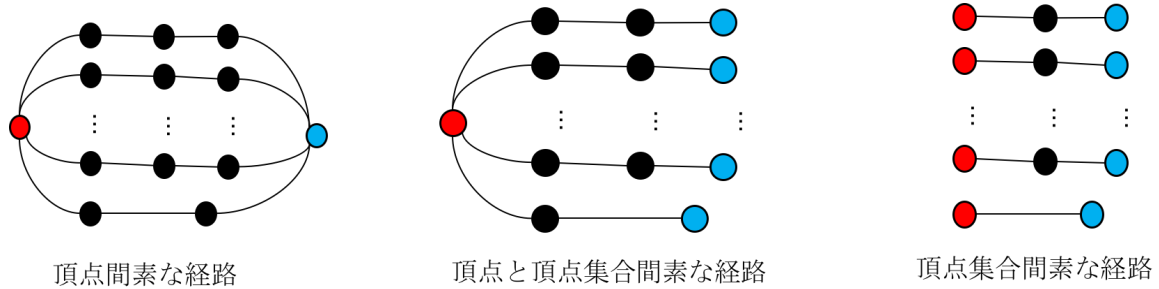


図12 素な経路の種類

## 7 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。

### 7.1 スターグラフ関連研究紹介

#### 7.1.1 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  $O(n \log n)$  ですが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。

### 7.1.2 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  $k$ -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。

### 7.1.3 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。

## 8 バブルソートグラフ

## 9 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。

### 9.1 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。

### 9.2 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 11 のような性質を持つ。

表 11  $n$ -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
$n!$	$n(n-1)/2$	$n(n-1)/2$	$n-1$

### 9.3 トランスポジショングラフの関連研究紹介

#### 9.3.1 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。

### 9.3.2 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。

## 10 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。

### 10.1 パンケーキグラフの関連研究紹介

#### 10.1.1 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。

#### 10.1.2 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。提案手法のキープポイントは  $P_n$  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。

$$\begin{aligned} C1 &= (1,2,3,4), (4,3,2,1), (2,3,4,1), (1,4,3,2), (3,4,1,2), (2,1,4,3), (4,1,2,3), (3,2,1,4) \\ C2 &= (2,1,3,4), (4,3,1,2), (1,3,4,2), (2,4,3,1), (3,4,2,1), (1,2,4,3), (4,2,1,3), (3,1,2,4) \\ C3 &= (1,3,2,4), (4,2,3,1), (3,2,4,1), (1,4,2,3), (2,4,1,3), (3,1,4,2), (4,1,3,2), (2,3,1,4) \end{aligned}$$

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。



## 11 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。

### 11.1 焦げたパンケーキグラフ

### 11.2 焦げたパンケーキグラフの関連研究紹介

#### 11.2.1 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。

$$\begin{aligned} C1 &= \{(1, 2, 3), (\bar{3}, \bar{2}, \bar{1}), (2, 3, \bar{1}), (1, \bar{3}, \bar{2}), (3, \bar{1}, \bar{2}), (2, 1, \bar{3}), \\ &\quad (\bar{1}, \bar{2}, \bar{3}), (3, 2, 1), (\bar{2}, \bar{3}, 1), (\bar{1}, 3, 2), (\bar{3}, 1, 2), (\bar{2}, \bar{1}, 3)\} \\ C2 &= \{(\bar{2}, 1, 3), (\bar{3}, \bar{1}, 2), (1, 3, 2), (\bar{2}, \bar{3}, \bar{1}), (3, 2, \bar{1}), (1, \bar{2}, \bar{3}), \\ &\quad (2, \bar{1}, \bar{3}), (3, 1, \bar{2}), (\bar{1}, \bar{3}, \bar{2}), (2, 3, 1), (\bar{3}, \bar{2}, 1), (\bar{1}, 2, 3)\} \\ C3 &= \{(\bar{1}, \bar{2}, 3), (\bar{3}, 2, 1), (\bar{2}, 3, 1), (\bar{1}, \bar{3}, 2), (3, 1, 2), (\bar{2}, \bar{1}, \bar{3}), \\ &\quad (1, 2, \bar{3}), (3, \bar{2}, \bar{1}), (2, \bar{3}, \bar{1}), (1, 3, \bar{2}), (\bar{3}, \bar{1}, \bar{2}), (2, 1, 3)\} \\ C4 &= \{(2, \bar{1}, 3), (\bar{3}, 1, \bar{2}), (\bar{1}, 3, \bar{2}), (2, \bar{3}, 1), (3, \bar{2}, 1), (\bar{1}, 2, \bar{3}), \\ &\quad (\bar{2}, 1, \bar{3}), (3, \bar{1}, 2), (1, \bar{3}, 2), (\bar{2}, 3, \bar{1}), (\bar{3}, 2, \bar{1}), (1, \bar{2}, 3)\} \end{aligned}$$

$BP_n$  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。

12 まとめ

13 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