

$n \geq 14$ パンケーキグラフの直径計算

鴻池 祐輔[†] 金子 敬一[†] 品野 勇治[†]

n パンケーキグラフは、 n 種類の記号で作られる順列をそれぞれ頂点とし、順列の前方反転によって移ることが可能な順列間を辺で結んだグラフである。 n パンケーキグラフは $n!$ 個の頂点を持つので、頂点数に対する多項式時間のアルゴリズムでは、 $n \geq 14$ パンケーキグラフでさえ解くことが困難な問題として知られている。実際に、これまでに求められている直径は、 $n = 13$ までである。本稿では、Heydari らが 13 パンケーキグラフの直径を求めるときに使った手法を基本として、不要な探索を行わないように発展させた手法を提案し、世界で初めて $n = 14, 15$ のパンケーキグラフの直径を与えた。

Computing the Diameter of $n \geq 14$ Pancake Graph

YUUSUKE KOUNOIKE[†], KEIICHI KANEKO[†] and YUUJI SHINANO[†]

The n -pancake graph is a graph whose vertices are labeled by permutations on n symbols. There is an edge between two vertices when the label of one derived from the other by some prefix reversal. Finding the diameter of the n -pancake graph is known to be a very hard problem, because n -pancake graph has $n!$ vertices. Actually, $n = 13$ has been the maximum size of the n -pancake graph which diameter was known so far. In this paper, we propose an extended method of the method developed by Heydari et al. They found the diameter of the 13-pancake graph by using their method. By using our method, we give previously unknown diameters of 14 and 15-pancake graphs.

1. はじめに

パンケーキ整列問題 (*pancake sorting problem*)

は 2) で提案された以下のような問題である。

The chef in our place is sloppy, and when he prepares a stack of pancakes they come out all different sizes. Therefore, when I deliver them to a customer, on the way to the table I rearrange them (so that the smallest winds up on top, and so on, down to the largest on the bottom) by grabbing several from the top and flipping them over, repeating this (varying the number I flip) as many times as necessary. If there are n pancakes, what is the maximum number of flips (as a function of n) that I will ever have to use to rearrange them?

この問題は、前方反転問題 (*prefix reversal problem*) とも呼ばれる。

パンケーキグラフ (*pancake graph*) は、1 から n までの記号で作られる順列をそれぞれ頂点とし、前方反転によって移ることが可能な順列の間を辺で結んだグラフである。 n によって異なるグラフが作れることから、 n パンケーキグラフ (n -pancake graph) と呼ぶ。 n パンケーキグラフは、 $n!$ 個の頂点を持つ、 $n - 1$ 次の正則グラフである。パンケーキ整列問題とパンケーキグラフの直径を求める問題は等価な問題で

ある。パンケーキグラフは、対称性、再帰性、次数と直径に対する頂点数の多さといった利点を持つことから、並列計算機システムにおける相互結合網として注目されている。このとき、グラフの直径は通信遅延を示す尺度の 1 つとなる。

n パンケーキグラフの直径を求めるには、ある頂点から全頂点への最短距離を求めれば良い。しかし、 n パンケーキグラフは $n!$ 個の頂点を持つので、ダイクストラ法などの頂点数や辺数に依存するアルゴリズムでは、計算時間と記憶容量が指数的に増加することとなり、すぐに現実的には解けなくなってしまう。本研究ではパンケーキグラフの再帰性に注目して、最短距離を求める必要がある頂点だけについて調べる手法を提案する。提案する手法は、Heydari ら⁶⁾ が $n = 13$ で直径を求めるときに使った手法を基本として、不要な探索を行わないように発展させた手法である。

これまで、13 パンケーキグラフまでの直径については既知であったが、 $n \geq 14$ のパンケーキグラフの直径は未知であった。本論文では、世界で初めて $n = 14, 15$ のパンケーキグラフの直径を与えた。

本論文は、次の構成で説明する。まず、第 2 章で問題を定式化する。その後、第 3 章で、既往研究と本論文との関連を明確にし、主たる結果の位置づけを示す。第 4 章では基本となっている Heydari らの手法を概観し、本論文での提案手法を示す。第 5 章では、直径の計算に要した計算時間、記憶容量などを示す。

[†] 東京農工大学工学部情報コミュニケーション工学科
Department of Computer, Information and Communication Sciences, Faculty of Engineering, Tokyo University of Agriculture and Technology

2. 問題の定式化

1 から n までの順列全体を集合 S_n として、与えられた n 枚のパンケーキの山を、一番上のものから順に、最小のパンケーキを 1, 最大を n とした順列 $\pi \in S_n$ で表す. 上から順に k 枚のパンケーキを反転した結果に対応する順列を σ とし, 順列 π を長さ $k, n-k$ の部分列 x, y によって $\pi = xy$ と表すと, $\sigma = x^R y$ となり, $\pi^k = \sigma$ と表す. ここで x^R は x の順序を反転したものとす. また, 続けて x_1 枚, x_2 枚と反転することを $\pi^{(x_1, x_2)}$ のように表す. さらに, π^x によって連続する x_1, x_2, \dots, x_m 枚の反転を表す. $\pi^x = e_n$ となるとき, x を π を整列する手順と呼ぶ. ここで, 順列 e_n は整列された状態の順列 ($12 \cdots n$) とする.

順列 π を整列する最短の回数を $f(\pi) = \min\{|x| : \pi^x = e_n\}$ とすると, パンケーキ整列問題は, $f(n) = \max\{f(\pi) : \pi \in S_n\}$ の値を求める問題となる.

$\pi \in S_n$ をそれぞれ頂点に割り当て, $\sigma = \pi^k$ となる頂点 π, σ の間を辺で結んだグラフをパンケーキグラフと呼ぶ. n によって異なるグラフが作れることから, それぞれのパンケーキグラフのことは n パンケーキグラフと呼び, P_n で表す. 例として 4 パンケーキグラフ (P_4) を図 1 に示す.

パンケーキグラフにおいて, 頂点に割り当てられた順列の記号を一定の規則で読みかえて作ったグラフは, 元のパンケーキグラフと同型である. この性質をパンケーキグラフの対称性と呼ぶ. P_n は, n 個の P_{n-1} から成る. これをパンケーキグラフの再帰性と呼ぶ. それぞれを部分パンケーキグラフと呼び, P_n^k で表す. P_n^k の各頂点の最後の記号はすべて k である.

グラフにおいて, 任意の 2 頂点間の最短距離 (頂点間を結ぶ辺の数) の最大をグラフの直径という. しかし, パンケーキグラフの対称性により, ある頂点からの最大の最短距離を調べれば充分である. e_n を割り当てた頂点を選ぶことで, n パンケーキグラフの直径を求めることは, パンケーキ問題における $f(n)$ を求めることと等価となる.

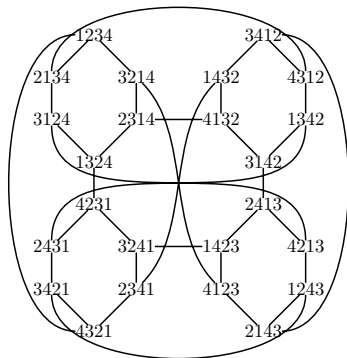


図 1 4 パンケーキグラフ
Fig. 1 The 4-pancake graph

3. 既往の研究との関連

パンケーキグラフの直径に関する研究としては, 直径の値を求めたもの, その上界・下界を示したのがある. これらの研究で求めた直径と, 本稿で世界で初めて求めた $14 \leq n \leq 15$ を合わせて, 表 1 に示す. また, それぞれの著者が直径を求めた n の範囲と, 直径の上界・下界を表 2 に示す.

また, Gates ら⁴⁾ は「焦げたパンケーキ問題 (*burnt pancake problem*)」という, それぞれのパンケーキの片面が焦げていると考え, 最終的にすべて焦げた面が下になるようにするという条件を追加した問題を提案している. この, 焦げたパンケーキ問題の性質から, 焦げていないパンケーキ問題の直径について, Cohen らと Heydari らがそれぞれ, $f(n) \leq (23/14)n + c$, $f(n) \leq 3(n+1)/2$ と予想している. 先に示した $f(14)$, $f(15)$ は, これらの予想を覆すものとはならなかった.

4. 直径計算の方法

4.1 Heydari らの手法

Heydari⁶⁾ らは, P_{13} のうち一部の頂点についてだけ最短距離を求めることで $f(13)$ を得る手法を提案している. P_n で部分パンケーキグラフ P_n^k を P_{n-1} とみなすと, 各頂点は P_{n-1} での整列の手順を使うことで, 同様に整列することができる. このうち, P_{n-1} で直径に等しい最短距離を持つ頂点 π それぞれに対し, $2 \leq k \leq n-1$ について, $\pi^{(n,k)}$ を列挙した頂点集合を S'_n とする. このとき, $\pi \in S'_n$ については $f(n-1)+2$ 回で, $\pi \in S_n \setminus S'_n$ については $f(n-1)+1$ 回以下でそれぞれ整列する手順を構築することが可能である. したがって, すべての $\pi \in S'_n$ について $f(\pi) \leq f(n-1)+1$ であれば, $f(n) \leq f(n-1)+1$ である. 1 つでも $f(\pi) = f(n-1)+1$ となれば, $f(n) = f(n-1)+1$ となる. S'_n は, S_n 全体に比べて小さいので, 効率良く $f(n)$ を得ることができる.

表 1 $f(n)$ の値
Table 1 The values of $f(n)$

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$f(n)$	0	1	3	4	5	7	8	9	10	11	13	14	15	16	17

表 2 既往の研究の成果
Table 2 The results of the previously works

著者	$f(n)$ の値	$f(n)$ の下界	$f(n)$ の上界
Garey ら ³⁾	$n \leq 7$	$n+1$	$2n-6$
Robbin ⁷⁾	$n \leq 9$		
Gates ら ⁴⁾			$(17/16)n$
Györi ら ⁵⁾			$5(n+1)/3$
Cohen ら ¹⁾	$n \leq 12$		$5(n+1)/3$
Heydari ら ⁶⁾	$n \leq 13$	$(15/14)n$	

4.2 本研究での提案手法

4.2.1 基本アルゴリズム

Heydari らの手法では P_{n-1} において最短距離が直径と等しいような頂点を列挙しておく必要がある．しかし，そのような頂点を列挙することは，直径を求めることと同様に困難である．6) ではその列挙について特に述べていないことから，すべての順列 $\pi \in S_{12}$ について $f(\pi)$ を計算していると思われる．

本稿で提案する手法では，この列挙を効率良く行うことで，Heydari らの手法よりもさらに少ない計算で $f(n)$ を得ることができる．

まず，順列 $\pi = e_{n-1}^x \in S_{n-1}$ に対して，順列 $\sigma_k \in S_n$ ($1 \leq k \leq n$) を式 (1) と定義すると， $f(\sigma_k)$ について式 (2) が成り立つ．

$$\sigma_k = \begin{cases} ((e_n)^n)^x & k = 1 \\ ((e_n)^{(k,n)})^x & 2 \leq k \leq n-1 \\ e_n^x & k = n \end{cases} \quad (1)$$

$$f(\sigma_k) \leq \begin{cases} f(\pi) + 1 & k = 1 \\ f(\pi) + 2 & 2 \leq k \leq n-1 \\ f(\pi) & k = n \end{cases} \quad (2)$$

順列 σ_k に対応する頂点は部分パンケーキグラフ P_n^k に含まれる．また， σ_k に対応する頂点と π に対応する頂点の関係はパンケーキグラフの再帰性と対称性から以下のように説明できる．

- 頂点 σ_n は， P_{n-1} の頂点 π に対応する．
- 頂点 σ_1 は， P_n^1 の記号の読みかえにより，頂点 $(e_n)^n$ が頂点 e_{n-1} に対応するようにしたときの，頂点 π に対応する．
- 頂点 σ_k ($2 \leq k \leq n$) は， P_n^k の記号の読みかえにより，頂点 $(e_n)^{(k,n)}$ が頂点 e_{n-1} に対応するようにしたときの，頂点 π に対応する．

$\pi \in S_{n-1}$ に対し， $\sigma_k \in S_n$ を求める変換を $T_k(\pi) = \sigma_k$ とし，集合 $S \subseteq S_{n-1}$ の要素すべてについて $T_k(\pi)$ を求めた集合を $T_k(S)$ とする．また，集合 S_n^m を $f(\pi) = m$ となる $\pi \in S_n$ の集合とし， $k < 0$ または $k > f(n)$ となる k については， S_n^k は空集合とする．このとき，集合 \overline{S}_n^m を式 (3) のように定義する．

$$\begin{aligned} \overline{S}_n^m &= T_1(S_{n-1}^{m-1}) \\ &\cup T_2(S_{n-1}^{m-2}) \cup \dots \cup T_{n-1}(S_{n-1}^{m-2}) \\ &\cup T_1(S_{n-1}^m) \end{aligned} \quad (3)$$

このとき，式 (2) により， $\pi \in \overline{S}_n^m$ について $f(\pi) \leq m$ が成り立つ．また， \overline{S}_n^m と， S_n^m, S_n の間には以下の関係が成り立つ．

$$S_n = \bigcup_{k=0}^{f(n-1)+2} \overline{S}_n^k \quad (4)$$

$$S_n^m \subseteq \bigcup_{k=m}^{f(n-1)+2} \overline{S}_n^k \quad (5)$$

すべての $\pi \in \overline{S}_n^{f(n-1)+2}$ について $f(\pi)$ を求めた結果， $f(\pi) = f(n-1) + 2$ となるような π があるか無

```

procedure make_problem( $n, m$ )
{ $S_n^k$  から  $\overline{S}_n^m$  を式 (3) により求める }
begin
  if  $n - 1 > c$  then begin
    make_problem( $n - 1, m - 2$ );
    solve( $n - 1, m - 2$ );
  end;
   $\overline{S}_n^m :=$  式 (3);
end

procedure solve( $n, m$ )
{ $\overline{S}_n^m$  を解き， $S_n^m$  に追加していく }
begin
  foreach  $\pi$  in  $\overline{S}_n^m$  do begin
     $f(\pi)$  を求める
     $S_n^{f(\pi)} := S_n^{f(\pi)} \cup \{\pi\}$ ;
  end
end

procedure diameter( $n$ )
{ $f(n)$  を求める }
begin
  make_problem( $n, f(n-1) + 2$ );
  solve( $n, f(n-1) + 2$ );
  if  $S_n^{f(n-1)+2}$  is not empty then
     $f(n) = f(n-1) + 2$ ;
  else
     $f(n) = f(n-1) + 1$ ;
end

```

図 2 $f(n)$ を求めるアルゴリズム

Fig. 2 Algorithm for computing $f(n)$

いかによって， $f(n), S_n^{f(n)}$ は以下ようになる．

- そのような π がある場合： $f(n) = f(n-1) + 2$ であり， $S_n^{f(n)}$ はすべて列挙されている．
- そのような π が無い場合： $f(n) \leq f(n-1) + 1$ であり， $f(\pi) = f(n-1) + 1$ となる π がある場合は $f(n) = f(n-1) + 1$ である． $S_n^{f(n)}$ を列挙するには，さらに $\overline{S}_n^{f(n-1)+1}$ について最短距離を求める必要がある．

後者の場合は，さらに $\overline{S}_n^{f(n-1)+1}$ を求める必要がある．さらに一般に $0 \leq m \leq f(n)$ とすると， S_n^m を完全に列挙するためには， \overline{S}_n^m が必要である． \overline{S}_n^m は $S_{n-1}^{m-2}, S_{n-1}^{m-1}, S_{n-1}^m$ から生成される．これらの集合は，再帰的に $S_{n-k}^{l \geq m-2k}$ を求めていくことで得ることができる． $S_1 = S_1^0 = \{e_n\}$ のように事前に列挙しておくことが容易なほど $n-k$ が小さくなるが， S_{n-k}^l が空集合となる $l < 0$ になるまで再帰を繰り返す．

図 2 にこのアルゴリズムを擬似コードで示す．ただし， S_c については事前に S_c^k ($k \leq f(c)$) を求めておくこととし， $\text{diameter}(c+1), \text{diameter}(c+2)$ と順番に呼び出すものとする． $c = 7$ としたとき，このアルゴリズムが S_n^m を求めていく順番を，表 3 に示す．

diameter 手続きを $c+1$ から昇順で呼び出すことで， $\text{solve}(n, m)$ が呼ばれる時点では， $k > m$ について $\text{solve}(n, k)$ を既に呼び出していることになる．したがって， $\text{solve}(n, m)$ 手続きが終了すると，式 (5) により S_n^m 全体が列挙されたことになる．

4.2.2 個々の最短距離の求め方

本研究では個々の最短距離の求め方として、A*探索による最短経路探索を用いた。A*探索の詳しい説明は8)などがあるので、ここでは省略する。

A*探索では、始点からある点までの距離とその点から終点までの距離の見積もりによって探索の順序を決定する。このとき、距離の見積もりが実際の距離の下界を与える場合、A*探索が見つける経路が最短であることが保証される。本研究では以下に示す $\tilde{f}(\pi)$ を距離の見積もりとして使用した。

$\pi \in S_n$ を、 $\pi = \pi_1\pi_2 \cdots \pi_n$ とする。 $\pi_{i+1} = \pi_i + 1$ または $\pi_{i+1} = \pi_i - 1$ のとき、 π_i と π_{i+1} が連結している、と呼ぶ。また、 $\pi_n = n$ のときも連結していることとし、 π の連結の数を π の連結度とする。このとき、1回の反転では連結度はたかだか1しか増えない。また、 e_n の連結度は n であることから、 $\tilde{f}(\pi)$ を $n - \pi$ の連結度とすると、 $\tilde{f}(\pi)$ は $f(\pi)$ の下界を与える。

4.2.3 実行の並列化

n が大きくなると、 \overline{S}_n^m は非常に多くの要素を持つようになるが、solve 手続きで行う最短経路の探索は、それぞれ完全に独立している。そこで、本研究では \overline{S}_n^m を適当に分割して、それぞれ別の計算機で実行するという単純な並列化を行った。分割の基準として、 \overline{S}_n^m が10億以上の要素を持つ場合に並列化を行うこととし、1000万個ずつに分割した。

5. 計算実験の結果

今まで未知であった $f(14)$, $f(15)$ を求められたことから、本研究で用いた手法は、十分に効果的であるといえる。 $c = 7$ からはじめ、 $f(n)$ を求めるまでにかかった時間と、最短距離を求めた問題数、 S_n の要素数を表4に示す。*印のない項目は Pentium 4 2.6CGHz, 1GB Dual Channel DDR-SDRAM で、*印のついた項目は上記に加え、Pentium3 1GHz Dual, 1GB PC133 ECC SDRAM の計算機16台を合わせて使用した。並列化の方式が単純で、負荷分散を行っていないことに加え、2種類のCPUにどのように割り振るかを調べるために少しずつ実行しているので、*印のついた項目の計算時間は目安程度に考えていただきたい。また、 S_n よりかなり少ない数で、 $f(n)$ が求められている。

表3 S_n^m を求める順番
Table 3 Computing order of S_n^m

n	f(n)	S_n^m を求める順番
8	9	$S_7^8 \rightarrow S_8^{10} = \phi$
9	10	$S_7^9 \rightarrow S_8^9 \rightarrow S_9^{11} = \phi$
10	11	$S_7^6 \rightarrow S_8^8 \rightarrow S_9^{10} \rightarrow S_{10}^{12} = \phi$
11	13	$S_7^5 \rightarrow S_8^7 \rightarrow S_9^9 \rightarrow S_{10}^{11} \rightarrow S_{11}^{13} \neq \phi$
12	14	$S_{11}^{13} \rightarrow S_{12}^{15} = \phi$
13	15	$S_7^4 \rightarrow S_8^6 \rightarrow S_9^8 \rightarrow S_{10}^{10}$ $\rightarrow S_{11}^{12} \rightarrow S_{12}^{14} \rightarrow S_{13}^{16} = \phi$

表4 $f(n)$ の計算にかかった時間と問題数

Table 4 Computing time and the number of problems for $f(n)$

n	計算時間	解いた問題数	$ S_n $
8	0.06 sec	210	40,320
9	1.0 sec	9,316	362,880
10	14.1 sec	117,996	3,628,800
11	3.5 min	1,425,037	39,916,800
12	0.07 sec	60	479,001,600
13	54 min	18,221,452	6,227,020,800
*14	8 hour	249,271,566	87,178,291,200
*15	7 day	3,640,943,222	1,307,674,368,000

6. おわりに

本稿では、世界で初めて14および15パンケーキグラフの直径を報告した。 $f(n)$ の下界に比べて上界は、第3章で示した予想が正しいとしても、まだ余裕がある。このことから、さらに上界を改善する余地があるのではないかと考えられる。

また、今後この手法を改良することにより、より大きな n について、パンケーキグラフと焦げたパンケーキグラフの両方の直径を求めたいと考えている。

最後に、本研究で発見した17回の反転が必要な長さ15の順列と、その整列の手順をいくつか示す。

- (1, 7, 3, 6, 4, 2, 8, 14, 12, 10, 13, 9, 11, 15, 5)
- (13, 4, 2, 5, 4, 2, 6, 13, 5, 2, 14, 15, 2, 4, 5, 7, 2)
- (1, 3, 2, 4, 15, 5, 13, 7, 10, 8, 11, 9, 12, 6, 14)
- (5, 15, 8, 2, 3, 4, 9, 3, 9, 5, 3, 8, 14, 4, 3, 2, 3)
- (1, 3, 15, 4, 13, 6, 8, 11, 9, 7, 10, 2, 12, 5, 14)
- (4, 2, 15, 14, 12, 5, 2, 7, 6, 5, 3, 10, 9, 7, 8, 13, 5)

参考文献

- 1) Cohen, D.S. and Blum, M.: On the problem of sorting burnt pancakes, *Discrete Appl. Math.*, Vol. 61, No. 2, pp. 105–120 (1995).
- 2) Dweighter, H.: *Amer. Math. Monthly*, Vol. 82, p. 1010 (1975).
- 3) Garey, M.R., Johnson, D.S. and Lin, S.: *Amer. Math. Monthly*, Vol. 84, p. 296 (1977).
- 4) Gates, W. H. and Papadimitriou, C. H.: Bounds for sorting by prefix reversals., *Discrete Math.*, Vol. 27, pp. 47–57 (1979).
- 5) Györi, E. and Turán, G.: Stack of Pancakes, Vol. 13, pp. 133–137 (1978).
- 6) Heydari, M. H. and Sudborough, I. H.: On the diameter of the pancake network, *J. Algorithms*, Vol. 25, No. 1, pp. 67–94 (1997).
- 7) Robbin, D. P.: personal communication (1977). cited in 4).
- 8) Russell, S. and Norvig, P.: エージェントアプローチ人工知能, 共立出版 (1997). 古川 康一 監訳.