A-032

焦げたパンケーキグラフにおけるコンテナ問題

Container Problem in Burnt Pancake Graphs

澤田直樹 * 鈴木康斗 * 金子敬一*
N. SAWADA Y. SUZUKI K. KANEKO

1 はじめに

本研究では，次数 n の焦げたパンケーキグラフ B_n[1] の任意の 2 節点に対して，n の多項式時間で，コンテナ問題を解く，すなわち n 本の内素な経路を作成するアルゴリズムを提案する．また，計算量のオーダおよび得られる経路長の最大値を見積もり，計算機実験により提案算法の性能評価を行う．

2 諸定義

定義 1 n 個の数 $a_i \in \{1, 2, \ldots, n\} (1 \le i \le n)$ および n 個の符号 $b_i \in \{-1, +1\} (1 \le i \le n)$ からなるような順列 $u = (a_1 \times b_1, a_2 \times b_2, \ldots, a_n \times b_n)$ を n 個の数 $1, 2, \ldots, n$ からなる符号付順列という．

定義 2 n 個の数 $1, 2, \ldots, n$ からなる任意の符号付順列 $u = (u_1, u_2, \ldots, u_n)$ に対し，前置反転操作 $u^{(i)} = (\overline{u_i}, \overline{u_{i-1}}, \ldots, \overline{u_1}, u_{i+1}, u_{i+2}, \ldots, u_n)$ を定義する．ただし，$-u_i$ は，$\overline{u_i}$ のように表記する．また，$u^{(i_1, i_2, \ldots, i_k)} = u^{(i_1, \ldots, i_{k-1})^{(i_k)}}$ と定義する．

定義 3 B_n は，$n! \times 2^n$ 個の節点を持つ．それぞれの節点は n 個の数 $1, 2, \ldots, n$ からなる符号付順列の 1 つを固有なラベルとして持つ．ラベル u を持つ節点は，集合 $\{u^{(i)} | 1 \le i \le n\}$ の要素をラベルとして持つ各節点とのみ隣接する．(図 1 参照)

B_n で，ラベルの最後を k に固定することで導出される部分グラフは，B_{n-1} となり，これを $B_{n-1}k$ で表す．B_n は，$2n$ 個の B_{n-1} からなる．B_n 中で 2 節点間の経路選択には，[2] の簡易算法を用いる．

* 東京農工大学工学部

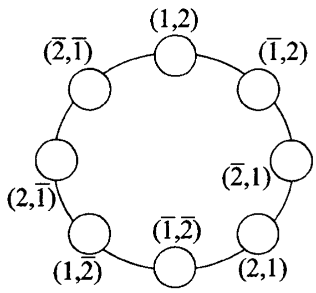

図 1: 次数 2 の焦げたパンケーキグラフ B_2

以下では，枝列 $(v_1, v_2), (v_2, v_3), \ldots, (v_{i-1}, v_i)$ からなる経路を，$v_1 \to v_2 \to \ldots \to v_i$ と表記し，簡易算法によって選択される経路を $\rightsquigarrow$ で表す．

3 算法

B_n で，任意の 2 節点 s, d 間の n 本の内素な経路を作成する算法を提案する．B_2 におけるコンテナ問題は自明であるため，以下では $n \ge 3$ と仮定する．

場合 I $s_n = d_n$

Step 1: $B_{n-1}s_n$ において，算法を再帰的に呼び出して，$s - d$ 間の内素な $n-1$ 本の経路を構成する．

Step 2: $s_1 \ne \overline{d_1}$ ならば，$s \to s^{(n)} \rightsquigarrow d^{(n)} \to d$，さもなくば，$s \to s^{(n)} \to s^{(n,1)} \rightsquigarrow d^{(n)} \to d$.

場合 II $s_n = \overline{d_n}$

Step 1: s からの経路 $p_1, p_2, \ldots, p_n$ を構成する．

$p'_i \ (1 \le i \le n-1) : s \to s^{(i)} \to s^{(i,n)}$.

$p_n : s \to s^{(n)}$.

Step 2: d からの経路 $q_1, q_2, \ldots, q_n$ を構成する．

$q_1 : d \to d^{(1)} \to d^{(1,n)}$.

$q'_j \ (2 \le j \le n-1) : j$ を $|d_j| = |s_i|$ を満たす整数とする．$d_j = s_i$ ならば，$d \to d^{(j)} \to d^{(j,n)}$，さもなくば，$d \to d^{(j)} \to d^{(j,1)} \to d^{(j,1,n)}$.

$q_n : d \to d^{(n)}$.

経路 p'_i と q'_j の終点を経由点とする．

Step 3: 経由点を 2 つ含む各部分グラフにおいて，2 節点間に，簡易算法で経路を構成する．

Step 4: 2 つの部分グラフが，経由点をただ 1 つ含むならば，その節点間に簡易算法で経路を構成する．

場合 III $|s_n| \neq |d_n|$

Step 1: s から d まで簡易算法で，経路 p を作る．また，d から s まで簡易算法で，経路 q を作る．p, q において，それぞれの d,s の隣接節点を $d^{(h)}, s^{(k)}$ とする．$s^{(k)} \in p$ ならば，$T = \{q\}$ とする．また，$d^{(h)} \in q$ ならば，$T = \{p\}$ とする．さもなければ，$T = \{p, q\}$ とする．

Step 2: s からの経路 $p_1, p_2, \ldots, p_n$ を構成する．

p_1 : 経路 T が節点 $s^{(1)}$ を含まないならば，$s \to s^{(1)} \to s^{(1,n)}$.

p'_i $(2 \leq i \leq n-1, i \neq k)$: 経路 T が節点 $s^{(i)}$ を含まないならば，$s \to s^{(i)} \to s^{(i,n)}$.

p_n : 経路 T が $s^{(n)}$ を含まないならば，$s \to s^{(n)}$.

Step 3: d からの経路 $q_1, q_2, \ldots, q_n$ を構成する．

q_1 : 経路 T が節点 $d^{(1)}$ を含まないならば，$d \to d^{(1)} \to d^{(1,n)}$.

q'_j $(2 \leq j \leq n-1, j \neq h)$: 経路 T が節点 $d^{(j)}$ を含むならば，経路 q_j を構成しない．j を $|d_j| = |s_i|$ を満たす整数とし，$d_j = s_i$ ならば，$d \to d^{(j)} \to d^{(j,n)}$，さもなくば，$d \to d^{(j)} \to d^{(j,1)} \to d^{(j,1,n)}$.

q_n : $d_1 \neq \overline{s_n}$ ならば，$d \to d^{(n)}$.

経路 p'_i と q'_j の終点を経由点とする．

Step 4: 経由点を 2 つ含む各部分グラフで，2 節点間に簡易算法で経路を構成する．

Step 5: 経由点をただ 1 つ含む部分グラフが，2 つか 4 つ存在する．それらの節点間に，内素となる経路を簡易算法で構成する．

4 計算量の推定と計算機実験

この算法の時間計算量，および選ばれる経路長の最大値は，それぞれ $O(n^3)$, $3n+4$ である．算法の能力を評価するために，B_n に対して，10,000 回ずつ無作為に s, d を定めて計算機実験を実行した．

図 2, 図 3 に，平均実行時間と 1 本の経路長の最大値を示す．これらより，平均時間計算量は $O(n^{2.6})$ であり，最大経路長が $3n+4$ であることが分かる．

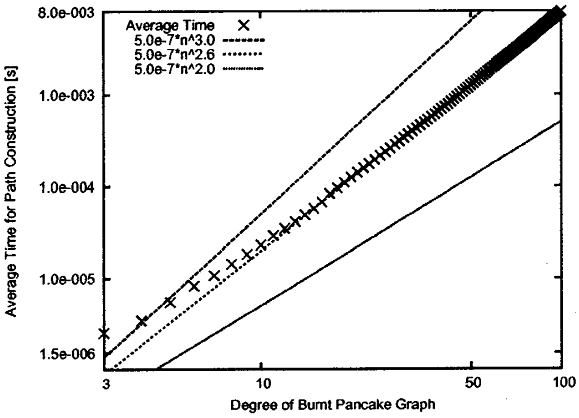

図 2: Average Time of Paths Construction

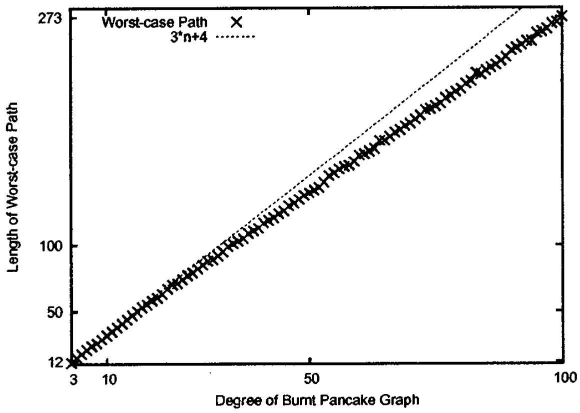

図 3: Length of Worst-case Path

5 結論

本研究では，B_n におけるコンテナ問題を，n の多項式時間で解く算法を提案した．その計算量，最大経路長は $O(n^3)$ および $3n+4$ となった．また，計算機実験により算法の性能評価を行い，平均計算量，経路長の最大値が，それぞれ $O(n^{2.6})$, $3n+4$ であることを示した．今後の課題として，他の相互結合網におけるコンテナ問題に対する算法の開発がある．

参考文献

[1] W.H.Gates et *al.*: Bounds for sorting by prefix reversal, *Discrete Math.*, 27(6), 47–57, 1979.

[2] K.Kaneko: An algorithm for node-to-set disjoint paths problem in burnt pancake graphs, *IEICE Trans. Info. & Sys.*, E86-D(12), 2588–2594, 2003.