Vol.2014-AL-148 No.17
2014/6/14

K-縮退グラフに含まれる誘導木の列挙

和佐 州洋[1]　有村 博紀[1]　宇野 毅明[2]

概要：本稿では，k-縮退グラフに含まれる誘導木（連結かつ非巡回な誘導部分グラフ）を，効率よく列挙するアルゴリズムを与える．グラフ $G = (V, E)$ が k-縮退 (k-degenerate) であるとは，G の任意の部分グラフが高々 k の次数を持つ頂点を含むときをいう．これまで，制約のある誘導部分グラフの列挙に関しては，連結な誘導グラフ (Avis, Fukuda, DAM, 1996) や，コードレスパス，コードレスサイクル (宇野, 第 92 回アルゴリズム研究会, 2003) の列挙に関する先行研究があるが，誘導木については知られていない．最大次数が d のとき，誘導木 1 つ当たり $O(d)$ 遅延の列挙アルゴリズムは容易に構成できるが，これをどの程度まで改善できるかは未解決の問題である．我々のアルゴリズムは，入力グラフ G が k-縮退グラフであるとき，誘導木を 1 つ当たりならし $O(k)$ 時間で列挙する．系として，k が定数ならば，誘導木を 1 つたり，ならし定数時間で列挙可能である．

Enumeration of Induced Subtrees in a K-Degenerate Graph

Kunihiro Wasa[1]　Hiroki Arimura[1]　Takeaki Uno[2]

1. 序論

グラフ中の部分構造列挙問題とは，部分グラフ族に関する制約条件が与えられた時に，入力グラフに含まれ，かつ，条件を満たすすべての部分構造を漏れなく重複なく列挙する問題である．部分構造列挙問題は，これまで広く研究されてきた [1–3, 6–9]．列挙アルゴリズムの研究では，遅延，つまり，解 1 つ当たりにかかる時間に着目して，研究が行われてきた．そこで，究極の目標は，定数時間遅延を持つアルゴリズムを与えることである．

これまでに，部分構造列挙について，次のような結果が示されている．以下では，入力グラフを $G = (V, E)$ と固定し，$m = |E|$ を G 中の辺の数，$n = |V|$ を G 中の頂点の数，N を解の個数とする．1970 年代に，Tarjan と Read [7] は，制約条件を全域木とした時の列挙問題を研究した．彼らは，G が与えられた時に，すべての全域木を $O(m + n + mN)$ 時間で列挙するアルゴリズムを与えた．1996 年に，Avis と Fukuda [1] は，G に含まれるすべての連結な誘導グラフを列挙する問題を研究した．彼らは，逆探索法を用いることで，これらを $O(mnN)$ 時間ですべて列挙するアルゴリズムを与えた．宇野 [9] は，G と頂点 $s, t \in V$ が与えられた時，s と t を結ぶすべてのコードレスパスを $O((m + n)N)$ 時間で，コードレスサイクルを $O((m + n)N)$ 時間で，すべて列挙するアルゴリズムを与えた．これらのアルゴリズムのならし時間遅延は，入力グラフのサイズの線形時間であることに注意されたい．

最近，解 1 つ当たりの遅延またはならし遅延を，列挙される解のサイズ程度，または，それ以下で抑える研究が登場してきた [3, 6, 8]．Shioura と，Tamura，Uno [6] は，すべての全域木を $O(n + N)$ 時間で列挙するアルゴリズムを

[1] 北海道大学大学院情報科学研究科
Hokkaido University, Graduate School of Information Science and Technology, {wasa, arim}@ist.hokudai.ac.jp
[2] 国立情報学研究所
National Institute of Informatics, uno@nii.jp

与えた．これは，前処理の時間を除き，解 1 つ当たりならし定数時間遅延で列挙するアルゴリズムである．Ferreira ら [3] は，制約条件を c-部分木，つまり，c 個の頂点からなる G 中の部分木とした時に，G 中のすべての c-部分木を $O(cN)$ 時間で列挙するアルゴリズムを与えた．これは，ならし $O(c)$ 時間遅延となる．Ferreira らの結果をさらに改善して，Wasa ら [8] は入力グラフを木に限定した時に，すべての c-部分木を，$O(n+N)$ 時間で列挙する最適なアルゴリズムを与えた．これは，ならし $O(1)$ 時間遅延となる．

入力グラフが木の場合には，その部分木は誘導グラフになるので，Wasa らの結果は，入力グラフが木の場合に，サイズ c の誘導部分木が定数時間遅延で列挙可能性なことを意味する．この結果より，木より一般的な入力グラフのクラスに対して，すべての誘導部分木が定数時間遅延で列挙可能かどうかは，自然な問いである．

本論文では，この問題に対して，入力グラフが k-縮退グラフであるときに，誘導木 1 つ当たりならし $O(k)$ 時間遅延で列挙するアルゴリズムを与える．これより，森や，格子グラフ，平面グラフなどの定数縮退数を持つ入力グラフのクラスに対して，すべての誘導木がならし定数時間遅延で列挙可能であることが示される．

本稿の構成は以下のとおりである．2 章では，本稿で使われる基本的な用語と問題に関する定義を与える．3 章では，入力グラフが k-縮退グラフであるときに，すべての誘導木をならし $O(k)$ 時間遅延で列挙するアルゴリズムを与える．4 章で，本稿のまとめと今後の課題を与える．

2. 準備

2.1 グラフ

グラフ (undirected graph) $G=(V(G),E(G))$ を，**頂点集合** $V(G)$ と**辺集合** $E(G) \subseteq V(G)^2$ の組で表す．頂点 $u \in V$ の**隣接頂点集合** $N_G(u)$ を，$N_G(u)=\{v \in V(G) \mid (u,v) \in E(G)\}$ とする．頂点 $u \in V$ の**次数**を $d_G(u)=|N_G(u)|$ とする．以下では，G は単純，つまり，多重辺と自己閉路を持たないとする．また，文脈から明らかならば，$V(G)$ と，$E(G)$，$N_G(u)$，$d_G(u)$ を，それぞれ，単に，V と，E，$N(u)$，$d(u)$ と書く．

G 中の任意の 2 頂点を $u,v \in V$ とし，j 個の頂点の列を $\pi=(v_1=u,\ldots,v_j=v)$ とする．また，i を $1 \le i \le j-1$ を満たす任意の整数とする．ここで，任意の π に対して，相異なる $j-1$ 個の辺 (v_i,v_{i+1}) が E に含まれるとき，π を u から v への**パス**といい，u から v へ**到達可能**であるという．路の中で，$j>2$ かつ $u=v$ であるとき π を**閉路**という．G が閉路を持たないとき，G を**非巡回**であるという．G 中の任意の相異なる 2 頂点間に路が存在するとき，G は**連結**であるという．

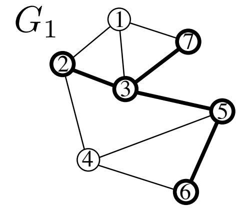

図 1　グラフ $G_1=(V_1,E_1)$ に含まれる誘導木 S_1 の例．図中の太線の頂点と辺は，誘導グラフ $G_1[S_1]$ 中の頂点と辺を示す．G_1 の頂点集合 V_1 は，集合 $\{1,\ldots,7\}$ からなり，辺集合 E_1 は，集合 $\{(1,2),(1,3),(1,7),\ldots,(5,6)\}$ からなる．また，S_1 は，集合 $\{2,3,5,6,7\}$ からなる．ここで，S_1 は連結かつ非巡回であるので，S_1 は誘導木である．

2.2 誘導木

頂点集合 S で誘導される G の**誘導グラフ** (induced subgraph) を，$G[S]=(S,E[S])$ と定義する．ただし，$E[S]=\{(u,v) \in E \mid u,v \in S\}$ とする．S のサイズを $|S|$ とする．誘導グラフ $G[S]$ は，頂点集合 S から一意に定まる．よって，以下では，文脈から明らかな場合，S と $G[S]$ を同一視する．S が非巡回であるとき，S を**誘導森** (induced forest) と呼び，さらに，誘導森 S が連結ならば，S を**誘導木** (induced subtree) と呼ぶ（図 1）．$G=(V,E)$ の**部分グラフ** (subgraph) を $G'=(V',E')$ とし，$V' \subseteq V$，$E' \subseteq E[V']$ とする．

S を G の誘導木とする．頂点 $x \in V$ と S の**隣接数**を $\mathrm{cnt}(S,x)=|\{y \in S \mid (x,y) \in E\}|$ とする．隣接数は，x と隣接する S 中の頂点の数を表す．ここで，隣接数に関して以下の補題が得られる．

補題 1. グラフ $G=(V,E)$ 中の誘導木を S とし，S に含まれない頂点を $u \in V \setminus S$ とする．このとき，$\mathrm{cnt}(S,u)>1$ ならば，$S \cup \{u\}$ は閉路を含む．

証明．頂点 u は，S との接続数が 2 以上であるので，S 中の相異なる頂点 v,w と隣接している．また，S は連結であるので，S 中で v から w までパスが存在する．よって，$S \cup \{u\}$ は，閉路 $c=(u,v,\ldots,w,u)$ を含む．■

2.3 *K*-縮退グラフ

グラフ G のすべての誘導グラフが，次数 k 以下の頂点を持つ時，G は k-**縮退グラフ** (k-degenerate graph) であるといい [4]，これを満たす最小の k を，G の**縮退数**と呼ぶ．

例 1. 定数の縮退数を持つグラフクラスとして，木や，格子グラフ，外平面グラフ，平面グラフが挙げられ，その縮退数は，それぞれ，1，および，2，2，5 である．

ここで，以下の補題が成り立つ．

補題 2. $G=(V,E)$ をグラフとし，k を整数とする．このとき，以下はすべて等価である．

(1) グラフ G のすべての誘導グラフが，次数 k 以下の頂点を持つ．

(2) グラフ G のすべての部分グラフが，次数 k 以下の頂点

を持つ．

(3) 条件 $\forall u \in V, |\{v \in V \mid v \in N(u), \sigma_G(u) < \sigma_G(v)\}| \leq k \ldots (*)$ を満たす全単射関数 $\sigma_G : V \to \{1, \ldots, |V|\}$ が存在する．

(4) 次数 k 以下の頂点とそれに隣接する辺を G から繰り返し除くことで，G が K_1 グラフになる．

証明．(1) → (2): グラフ G のすべての誘導グラフが，次数 k 以下の頂点を持つとする．ここで，G 中のある部分グラフを $G' = (V', E')$ とすると，$G[V']$ の辺集合 $E[V']$ は，$E' \subseteq E[V']$ である．よって，$G[V']$ は次数 k 以下の頂点を持つことから，S は次数 k 以下の頂点を持つ．(2) → (3): グラフ $G = (V, E)$ のすべての部分グラフが，次数 k 以下の頂点を持つとする．このとき，$G_1 = G$ は k 以下の頂点を持つことから，その頂点を u_1 とし，$\sigma_G(u_1) = 1$ とする．つぎに，$G_2 = (V \setminus \{u_1\}, E[V \setminus \{u_1\}])$ は，同様に次数 k 以下の頂点を持ち，その頂点を u_2 とし，$\sigma_G(u_2) = 2$ とする．これを繰り返すことで，条件 (*) を満たす σ を構成できる．(3) → (4): 条件 (*) を満たす σ を，G に与えたと仮定する．このとき，σ に関して，小さい頂点から順に G から除くことで，K_1 が得られる．(4) → (1): 次数 k 以下の頂点とそれに隣接する辺を G から繰り返し除くとし，最初に除く頂点を u_1，次に除く頂点を u_2 とし，最後に残った頂点を $u_{|V|}$ とする．ここで，任意の $1 \leq i \leq |V|$ に対して，G_i を $V_i = \{u_i, \ldots, u_{|V|}\}$ で誘導されるグラフ $G_i = G[V_i]$ とする．このとき，u_i を含む任意の頂点集合 U から誘導されるグラフ $G_i[U]$ において，u_i は $G_i[U]$ 中で，次数が高々 k の頂点である．よって，グラフ G のすべての誘導グラフは，次数 k 以下の頂点を持つ． ■

補題 2 の (3) の条件 (*) を満たす σ_G の例を図 2 に与える．ここで，$u < v (u > v)$，つまり，頂点 $u \in V$ が頂点 $v \in V$ より小さい（大きい）とは，$\sigma_G(u) < \sigma_G(v) (\sigma_G(u) > \sigma_G(v))$ を満たすときをいい，全単射関数 σ_G を G の**順序**という．以下では，文脈から明らかならば，σ_G を単に σ と書く．

補題 2 の (3) は，グラフが k-縮退グラフならば，任意の頂点 $u \in V$ に対して，u の隣接頂点集合 $N(u)$ 中で，u よりも大きい頂点の数が高々 k であるような順序が存在することを示している．任意の k-縮退グラフ G に対して，G の順序 σ を与える線形時間アルゴリズム [5] が，Matula と Beck によって与えられている．さらに，Matula と Beck のアルゴリズムを用いて，G の縮退数を得ることができる．

2.4 列挙アルゴリズム

本節では，列挙アルゴリズムの定義を与える．列挙問題 Π に対する列挙アルゴリズムとは，インスタンス I を受け取り，解集合 $S(I)$ に含まれるすべての解 S を漏れ無く重複無く出力するアルゴリズム A をいう．本稿で扱う問題を，以下に定義する．

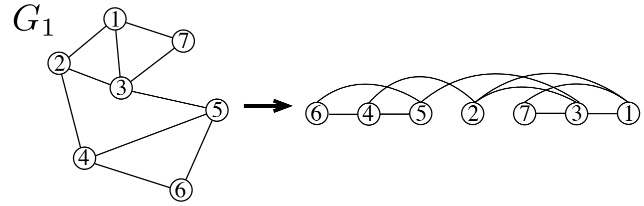

図 2 グラフ $G_1 = (V_1, E_1)$ に対して，順序 σ_1 を与えた例．矢印を挟んで左右のグラフは同型である．右のグラフでは，頂点が左から右に昇順で並んでいる．すなわち，$\sigma_1(6) = 1, \sigma_1(4) = 2, \ldots, \sigma_1(1) = 7$ である．また，G_1 は 2-縮退グラフであり，σ_1 は補題 2 の (3) の条件 (*) を満たしている．

問題 1. $G = (V, E)$ を入力グラフとする．このとき，誘導木をなす頂点集合 $S \subseteq V$ をすべて列挙せよ．

ここで，$n = |I|$ とし，$N = |S(I)|$ とする．このとき，任意の多項式 $p(\cdot)$ に対して，アルゴリズム A が**ならし多項式遅延列挙アルゴリズム**であるとは，入力 I を受け取り，A が停止するまでの時間計算量が $O(N \cdot p(n))$ であるようなアルゴリズムをいう．また，$O(p(n))$ を**ならし多項式時間遅延**という．

3. 提案アルゴリズム

本章では，入力グラフ $G = (V, E)$ が k-縮退グラフであるときに，G に含まれる誘導木を，ならし $O(k)$ 時間遅延で列挙する，分割法をベースにしたアルゴリズムを与える．

3.1 候補集合

本節では，誘導木に追加しても誘導木の性質を保つ頂点の集合である，候補集合の関する定義を与える．まず，以下に，候補集合の定義を与える．

定義 1. グラフ $G = (V, E)$ 中の誘導木を S とする．このとき，**候補集合** $Can(S)$ を以下のように定義する：

$$Can(S) = \{u \in V \setminus S \mid \mathrm{cnt}(S, u) = 1\}.$$

補題 3. グラフ $G = (V, E)$ 中の誘導木を S とし，頂点 u を S に含まれない頂点とする．ここで，頂点 u が S の候補集合に含まれるならば，その時に限り，$S \cup \{u\}$ は誘導木である．

証明．まず，十分条件を示す．$u \in Can(S)$ と仮定する．このとき，$\mathrm{cnt}(S, u) = 1$ であることから，$S \cup \{u\}$ は連結であり，また，少なくとも u を含む閉路を含まない．よって，S も閉路を含まないことから，$S \cup \{u\}$ は誘導木である．次に，必要条件を示す．$S \cup \{u\}$ が誘導木であると仮定する．このとき，u は $S \cup \{u\}$ 中で，次数 1 以上である．一方，u の次数が 2 以上とする．このとき，$S \cup \{u\}$ は u を通るパス π を持ち，π の両端の頂点を s, t とする．ここで，$S \cup \{u\}$ が誘導木であることから，s から t までのパスは，π のみである．よって，S において，s から t へは到達可能ではなく，S は非連結である．したがって，$\mathrm{cnt}(S, u) = 1$

図 3 手続き ISE($G=(V,E)$)：G 中のすべての誘導木を列挙する主手続き．

Step I1. 補題 2 の (3) の (*) を満たす順序を計算する．
Step I2. $\emptyset$ を出力する．
Step I3. u を V 中の最小の頂点とする．
Step I4. S を $\{u\}$ とし，$Can(S)$ を求める．
Step I5. $\textsc{Rec}(S, Can(S))$ を呼び出す．
Step I6. G から u を除く．
Step I7. G が空でなければ，Step I3 に戻る．

である． ■

補題 3 は，サイズ $i+1>0$ の任意の誘導木を得るには，サイズ i のある誘導木 S に，候補集合 $Can(S)$ 中の頂点を加える事が，必要十分条件であることを示している．

3.2 アルゴリズム

本節では，入力グラフ $G=(V,E)$ に含まれるすべての誘導木を列挙する提案アルゴリズムを与える．

提案アルゴリズムの概要を以下に述べる．V の部分集合をすべて含む集合族を $\mathcal{S}(V)$ とし，$\mathcal{S}(V)$ を**探索空間**と呼ぶ．ここで，V 中の頂点 v を含む $\mathcal{S}(V)$ の部分探索空間を $\mathcal{S}_{v^+}(V)$ とし，含まない部分探索空間を $\mathcal{S}_{v^-}(V)$ とする．提案アルゴリズムは，探索空間 $\mathcal{S}$ を，部分探索空間 $\mathcal{S}_{v^+}(V)$ と $\mathcal{S}_{v^-}(V)$ の互いに素な空間へ再帰的に分割しながら，部分探索空間を探索することで，誘導木を列挙するアルゴリズムである．

提案アルゴリズムは，候補集合を用いることで，探索空間を枝刈りし，効率よく列挙するアルゴリズムである．具体的には，誘導木 S の候補集合が空である場合，補題 3 より，$V\setminus S$ 中のいかなる頂点 u を加えても，$S\cup\{u\}$ は閉路を持つ．したがって，提案アルゴリズムは，候補集合が空になるまで，候補集合中の頂点を追加し，空になったとき，それ以降の分割を行わず，他の部分探索空間を探索することで効率化を図る．

提案アルゴリズムのメインルーチン ISE とその手続き Rec の擬似コードを，図 3 と図 4 に与える．手続き ISE($G=(V,E)$) は，入力として無向グラフ $G=(V,E)$ を受け取る．ISE は，まず，空な誘導木を出力する．次に，サイズ 1 からなる誘導木とその候補集合を入力とする再帰手続き Rec を呼び出し，それをすべての頂点について行うことで，G に含まれるすべての誘導木を出力する．

手続き $\textsc{Rec}(S, Can(S))$ は，入力として，G 中の誘導木 S と S の候補集合 $Can(S)$ を受け取る．Rec は，まず，S を出力する．次に，S よりサイズが 1 つ大きい誘導木を出力するために，候補集合中の頂点 u を加えた誘導木 $S\cup\{u\}$ とその候補集合を入力とする Rec を，再帰的に呼び出す．これをすべての候補集合中の頂点に対して行う．

次に，提案アルゴリズムの正当性に関する定理を与える．

定理 1. 提案アルゴリズムは入力グラフ G 中の誘導木すべてを，漏れなく重複なく列挙する．

図 4 手続き $\textsc{Rec}(S, Can(S))$

Step R1. S を出力する．
Step R2. u を $Can(S)$ 中の最小の頂点とする．
Step R3. $Can(S\cup\{u\})$ を求める．
Step R4. $\textsc{Rec}(S\cup\{u\}, Can(S))$ を呼び出す．
Step R5. G と $Can(S)$ から u を除く．
Step R6. $Can(S)$ が空でなければ，Step R2 に戻る．

証明． (I) まず，アルゴリズムが誘導木を漏れなく出力することを，出力される誘導木のサイズに関する帰納法を用いて示す．サイズが 0 の誘導木は，Step I2 で出力される．次に，サイズが 1 の誘導木について考察する．Step I5 で，V 中の一つの頂点 u からなる集合 S と $Can(S)$ を引数として，Rec を呼び出し，Step R1 で S を出力している．これを V 中のすべての頂点に関して行っているので，ISE は，サイズ 1 のすべての誘導木を出力する．次に，サイズ i の誘導木 S がすべて出力されたと仮定する．このとき，アルゴリズムは，S を出力した後，S に $Can(S)$ 中の頂点一つを加えた誘導木 S' と S' の候補集合を引数として，Rec を再帰的に呼び出し，これを $Can(S)$ 中の頂点全てに対して行う．また，補題 3 より，サイズ $i+1$ の任意の誘導木は，サイズ i のある誘導木に候補集合中の頂点を追加することで，生成できる．よって，仮定より，サイズ i の誘導木はすべて出力されるため，サイズ $i+1$ の誘導木はすべて出力される．以上の議論から，アルゴリズムは，G 中の誘導木を漏れなく出力する．

(II) 次に，出力に重複がないことを，数学的帰納法を用いて示す．アルゴリズムは，明らかに，サイズ 0 とサイズ 1 の誘導木を重複なく出力する．また，サイズ $i\geq 1$ までの誘導木が重複なく出力されたと仮定する．ここで，サイズ $i+1$ の誘導木 $S\cup\{u,v\}$ が出力される回数について考察する．ただし，S をサイズ $i-1$ の誘導木とし，u,v を $u<v$ を満たす $V\setminus S$ 中の頂点とする．

このとき，次の 2 つの場合がある．(II.a) まず，$u\in N(v)$ のとき，u または v のいずれかが $Can(S)$ に含まれ，もう一方が $Can(S\cup\{w\})$ に含まれる．ただし，w は，u,v のうち，$Can(S)$ に含まれる頂点とする．よって，追加される順番は，一意に決まる．(II.b) 次に，$u\neq N(v)$ のとき，u,v ともに，$Can(S)$ に含まれる．このとき，u より先に v を追加すると，u は v よりも小さいため，Step R5 で，u は G から除かれている．よって，アルゴリズムは，u,v ともに S に追加するとき，必ず u から先に追加する．したがって，(II.a) と (II.b) より，u と v ともに S に追加される順番は，一意であり，サイズ $i+1$ の誘導木は一度しか出力されない．(I) と (II) より，題意は示された． ■

3.3 計算量の解析

本節では，提案アルゴリズムの計算量を解析する．まず，

$Can(S)$ と $Can(S \cup \{u\})$ との差分から，$Can(S \cup \{u\})$ を計算する方法を，次に与える．

補題 4. グラフ $G = (V, E)$ 中の誘導木を S とし，頂点 u を S の候補集合 $Can(S)$ 中の頂点とする．このとき，以下が成り立つ：

$$Can(S \cup \{u\}) = (Can(S) \cup \Gamma) \setminus \Delta.$$

ただし，$\Gamma = \{v \in N(u) \mid \mathrm{cnt}(S, v) = 0\}$ とし，$\Delta = (\{v \in N(u) \mid \mathrm{cnt}(S, v) > 0\} \cup \{u\})$ とする．

証明. (I) $Can(S \cup \{u\}) \subseteq (Can(S) \cup \Gamma) \setminus \Delta$ を示す．w を $Can(S \cup \{u\})$ 中の任意の頂点とする．このとき，定義より，$\mathrm{cnt}(S \cup \{u\}, w) = 1$ であるので，w は，$S \cup \{u\}$ 中で u と u 以外のただひとつの頂点 x の，いずれか一方と隣接している．よって，$w \notin \Delta$ である．また，$w \in N(u)$ のとき，$w \in \Gamma$ であり，さらに，$w \notin N(u)$ のとき，$w \in Can(S)$ である．よって，$w \in (Can(S) \cup \Gamma) \setminus \Delta$ であるので，$Can(S \cup \{u\}) \subseteq (Can(S) \cup \Gamma) \setminus \Delta$ である．

(II) $Can(S \cup \{u\}) \supseteq (Can(S) \cup \Gamma) \setminus \Delta$ を示す．w を $(Can(S) \cup \Gamma) \setminus \Delta$ 中の任意の頂点とする．まず，$u \in \Delta$ であるので，$w \neq u$ である．また，$w \notin S$ である．次に，w と $S \cup \{u\}$ との接続数を考える．ここで，$Can(S)$ と Γ は互いに素であるので，以下のように場合分けする．(II.a) $w \in Can(S)$ のとき，w は，S 中のただひとつの頂点と隣接している．さらに，$N(u)$ と隣接し，かつ S 中の頂点と隣接している頂点は Δ に含まれるので，$\mathrm{cnt}(S \cup \{u\}, w) = 1$ である．(II.b) $w \in \Gamma$ であるとき，w は，$S \cup \{u\}$ 中で u のみと隣接する．よって，(II.a) と (II.b) より，$\mathrm{cnt}(S \cup \{u\}, w) = 1$ である．(I) と (II) の議論から，題意は示された． ■

次に，頂点 u の回避集合に関する定義を与える．

定義 2. グラフ $G = (V, E)$ 中の誘導木 S と頂点 $u \in V$ に対して，u の**回避集合** $Avo_S(u)$ を以下のように定義する：

$$Avo_S(u) = \{v \in N(u) \mid v < u, \mathrm{cnt}(S, v) \geq 1\}.$$

ただし，頂点の大小は，σ によって定まる順序とする．

誘導木 S における回避集合族を $Avo(S) = \bigcup_{u \in V} Avo_S(u)$ とする．回避集合に関して，補題 1 より，以下の補題が直ちに得られる．

補題 5. G 中の誘導木を S とし，候補集合 $Can(S)$ 中の頂点を u とする．このとき，v が u の回避集合 $Avo_S(u)$ 中の頂点ならば，$v \notin Can(S \cup \{u\})$ である．

次に，候補集合を更新するための ISE の手続きである UpdateCandidate とその副手続き UpdateAvoid の擬似コードを図 5 と 6 に与える．それぞれの詳細を与える．

手続き UpdateCandidate$(S, Can(S), Avo(S), u = \min Can(S))$ は，入力として誘導木 S と，その候補集合 $Can(S)$，回避集合族 $Avo(S)$，S に加える候補集合中で最小の頂点 $u = \min Can(S)$ を受け取る．UpdateCandidate は，u の回避集合以外の隣接頂点に対して，S との隣接数で場合分けをしながら，$Can(S)$ と $Can(S \cup \{u\})$ の差分を計算することで，候補集合を更新する．

手続き UpdateAvoid(v, u) は，UpdateAvoid は，入力として頂点 v と追加頂点 u を受け取り，v の隣接頂点すべてに対して，その回避集合を更新する手続きである．

図 5 手続き UpdateCandidate$(S, Can(S), Avo(S), u = \min Can(S))$

Step C1. $Can(S \cup \{u\}) \leftarrow Can(S) \setminus \{u\}$ とし，$Avo(S \cup \{u\}) \leftarrow Avo(S)$ とする．
Step C2. $N(u) \setminus Avo_S(u)$ 中の各頂点 v に対して，Step C3 と Step C4 を繰り返す．
Step C3. $\mathrm{cnt}(S, v) = 0$ なら，$Can(S \cup \{u\})$ に v を加え，UpdateAvoid(v, u) を呼び出す．
Step C4. $\mathrm{cnt}(S, v) > 0$ なら，$Can(S \cup \{u\})$ から v を除く．
Step C5. $Can(S \cup \{u\})$ を返す．

図 6 手続き UpdateAvoid(v, u)

Step A1. $N(v)$ 中の各頂点 w に対して, $v < w$ ならば, $Avo_{S \cup \{u\}}(w)$ に v を加える．

補題 6. UpdateCandidate は，入力を誘導木 S と，その候補集合 $Can(S)$，回避集合族 $Avo(S)$，候補集合中の最小の頂点 $u = \min Can(S)$ としたとき，$S \cup \{u\}$ の候補集合 $Can(S \cup \{u\})$ を正しく返し，さらに回避集合を正しく更新する．

証明. (I) まず, UpdateCandidate が, 補題 4 の $Can(S \cup \{u\}) = (Can(S) \cup \Gamma) \setminus \Delta$ に対応しているか考察する．u が, $Can(S)$ 中で最小であることから, $Avo_S(u)$ は, $Can(S)$ 中の頂点を含まない．さらに，$Avo_S(u)$ 中の頂点は，補題 5 より，$Can(S \cup \{u\})$ 中に含まれない．よって，これらから，$Avo_S(u)$ と Γ，$Avo_S(u)$ と Δ はそれぞれ互いに素である．したがって，Step C3 で追加される頂点の集合は，Γ に対応し，Step C4 で削除される頂点の集合は，Δ に対応しているので，候補集合を正しく更新する．(II) 次に，回避集合については考察する．ある頂点 w の回避集合 $Avo_{S \cup \{u\}}(w)$ に頂点 v が追加されるのは，w より小さい w の隣接頂点で，かつその接続数が，$\mathrm{cnt}(S, v) = 0$ から $\mathrm{cnt}(S \cup \{u\}, v) = 1$ になる時である．このような頂点は，Step C3 の条件を満たす頂点 v であるので，Step C3 で呼び出される UpdateAvoid で，正しく更新される．よって，(I) と (II) から，題意は示された． ■

提案アルゴリズムは, 手続き UpdateCandidate と UpdateAvoid を用いて，候補集合の計算を行う．次に，本論文の主定理を述べる．

定理 2. 提案アルゴリズムは，入力グラフ $G = (V, E)$ 中に含まれるすべての誘導木を，前処理に $O(|V| + |E|)$ 時間，解 1 つ当たりならし $O(k)$ 時間遅延で，漏れなく重複なく列挙する．

証明. 提案アルゴリズムの正当性は，定理 1 と補題 6 から明らかである．以下では，時間計算量について議論する．アルゴリズムは，まず，順序を得るために，Matula と Beck [5] のアルゴリズムを用いることで，$O(|V|+|E|)$ 時間かかる．また，すべての頂点に関して，回避集合を空集合に初期化する必要がある．よって，前処理にかかる時間は，$O(|V|+|E|)$ 時間である．

次に，誘導木 1 つ当たりにかかる時間を考察する．Step I2 と Step I3 にかかる時間は，$O(1)$ 時間である．また，Step I4 にかかる時間は，$O(1)$ 時間である．Step I6 は，u を実際に G から削除すると，$O(d(u))$ かかってしまうが，擬似的に u の接続数を 1 にして $\textsc{UpdateAvoid}(u)$ を実行することで，アルゴリズムの正当性を失わずに，G から削除したときと同じ処理を実現できる．Step I7 は，$O(1)$ 時間で判定できる．

最後に, Step I4 にかかる時間, つまり, 候補集合の更新手続き $\textsc{UpdateCandidate}$ にかかる時間を考察する．Step C1 は，実際にはコピーせず，また，集合を双方向連結リストで実装することで，$O(1)$ 時間で実行できる．Step C2 にかかる時間は，$O(1)$ 時間である．次に，$Can(S \cup \{u\})$ のサイズを c とする．ここで，$\textsc{UpdateAvoid}(v)$ は，v の隣接頂点で，かつ v より大きい頂点の数は，高々 k 個であるから，$O(k)$ 時間で実行できる．よって，Step C3 にかかる時間は，$O(ck)$ 時間である．また，Step C4 にかかる時間は，この条件を満たす頂点はすべて u より大きいため，$O(k)$ 時間である．したがって，誘導木 1 つ当たり $O(ck)$ 時間かかる．ここで，この誘導木に候補集合中の頂点を加える事で生成される誘導木の数が，c 個であることから，これらの誘導木に $O(k)$ 時間ずつ時間を割り振るとする．このとき，各誘導木に割り当てられる時間計算量は，アルゴリズム全体を通して，高々 1 回である．

よって，誘導木 1 つ当たりにかかる時間は $O(k)$ 時間である．ここまでの議論を，誘導木の出力を差分で出力するとすると，同様の議論を $\textsc{Rec}$ に関しても行えるので，提案アルゴリズムは，入力グラフ $G=(V,E)$ 中に含まれるすべての誘導木を，$O(|V|+|E|)$ 時間の前処理のあとに，ならし $O(k)$ 時間遅延で，漏れなく重複なく列挙する． ■

定理 2 より，次の系が直ちに導かれる．

系 1. 入力グラフ $G=(V,E)$ が定数縮退グラフであるとする．このとき，G に含まれるすべての誘導木は，ならし定数時間遅延で列挙可能である．

4. 結論

本稿では，入力グラフ G が k-縮退グラフ中であるときに，G に含まれるすべての誘導木を，ならし $O(k)$ 時間遅延で列挙するアルゴリズムを提案した．これにより，定数縮退数を持つ入力グラフのクラスに対して，すべての誘導木がならし定数時間遅延で列挙可能であることが示された．

今後の課題として，ならしではなく，厳密に $O(k)$ 時間遅延で列挙できるアルゴリズムが存在するかどうかが，未解決問題である．また，入力グラフが k-縮退グラフであるときに，他の部分グラフを効率よく列挙する方法があるか，が挙げられる．

謝辞

本研究の一部は，文科省科学研究費補助金 特別研究員奨励費 25 · 1149，および，同基盤研究（A）24240021 の助成を受けた．

参考文献

[1] Avis, D. and Fukuda, K.: Reverse search for enumeration. *DAM*, 65:21–46, 1996.
[2] Birmelé, E., Ferreira R. A., Grossi R., Marino A., Pisanti N., Rizzi R., and Sacomoto G.: Optimal Listing of Cycles and st-Paths in Undirected Graphs. In *Proc. SODA 2013*, 1884–1896, 2013.
[3] Ferreira, R. , Grossi, R. , and Rizzi, R.: Output-sensitive listing of bounded-size trees in undirected graphs. In *Proc. ESA 2011*, LNCS 6942, 275–286, 2011.
[4] Lick, D. R. and White, A. T,: k-DEGENERATE GRAPHS. *Can. J. Math.*, XXII(5): 1082–1096, 1970.
[5] Matula, D. W. and Beck, L. L.: Smallest-last ordering and clustering and graph coloring algorithms. *J. ACM*, 30(3): 417–427, 1983.
[6] Shioura, A., Tamura, A., and Uno, T.: An optimal algorithm for scanning all spanning trees of undirected graphs. *SIAM J. Comput.*, 26(3):678–692, 1997.
[7] Tarjan, R. E. and Read, R. C. .: Bounds on backtrack algorithms for listing cycles, paths, and spanning trees. *Networks*, 5(3):237–252, 1975.
[8] Wasa, K., Kaneta, Y., Uno, T., and Arimura, H.: Constant time enumeration of bounded-size subtrees in trees and its application. In *Proc. COCOON 2012*, LNCS 7434, 2012.
[9] 宇野毅明: コードレスサイクルを列挙する線形時間アルゴリズム．第 92 回アルゴリズム研究会，岐阜市，2003.