グラフの幅2の真のパス分解を求める効率的アルゴリズム

松林昭，上野修一

東京工業大学電気・電子工学科
東京都目黒区大岡山 2-12-1

与えられた任意のグラフ G と整数 k に対して，G の真のパス幅が k 以下であるか否かを判定する問題は NP 完全であるが，k を定数に固定した場合には P に属すことが知られている．特に，k を 2 に固定した場合には，この問題を解く多項式時間アルゴリズムが明示されているが，その計算時間には天文学的に大きな定数係数が含まれており，現実的な意味で効率的であるとはいえないし，真のパス分解を与えないという意味で構成的でもない．本稿では，最大次数が 3 以下であるグラフ G の真のパス幅が 2 以下であるための必要十分条件を示し，これに基づいて G の幅 2 の真のパス分解を求める実際的な線形時間アルゴリズムを示す．

An Efficient Algorithm for Computing Proper-Path-Decomposition of Width Two

Akira Matsubayashi and Shuichi Ueno

Department of Electrical and Electronic Engineering
Tokyo Institute of Technology
2-12-1 O-okayama, Meguro-ku, Tokyo 152, Japan

It is known that the problem of determining, given a graph G and an integer k, whether G has proper-path-width at most k is NP-complete, while it is in P if k is a fixed integer. Although a polynomial time algorithm to solve the problem is known for a special case of $k = 2$, the algorithm is neither practical nor constructive since the time complexity involves an enormous constant factor and the algorithm provide no proper-path-decomposition. In this paper, we show a necessary and sufficient condition for a graph with maximum vertex degree 3 to have proper-path-width 2, and based on the condition, we give a practical linear time algorithm for computing a proper-path-decomposition of width at most 2.

1 まえがき

グラフ G の点集合と辺集合をそれぞれ $V(G)$ と $E(G)$ で表す．グラフ G に対して，$V(G)$ の部分集合の系列 $\mathcal{X} = (X_1, \ldots, X_r)$ が以下の条件をすべて満たすとき，$\mathcal{X}$を G の真のパス分解といい，$\max_{1 \leq i \leq r} |X_i| - 1$ を $\mathcal{X}$ の幅という：

(i) $X_i \not\subseteq X_j \ (i \neq j)$;

(ii) $\bigcup_{1 \leq i \leq r} X_i = V(G)$;

(iii) $\forall (u,v) \in E(G) \ \exists i \text{ s.t. } u, v \in X_i$;

(iv) $X_a \cap X_c \subseteq X_b \ (1 \leq a \leq b \leq c \leq r)$;

(v) $|X_a \cap X_c| \leq |X_b| - 2 (1 \leq a < b < c \leq r)$.

G の真のパス分解の幅の最小値を G の真のパス幅といい，$ppw(G)$ で表す [3]．G の真のパス分解は幅 $ppw(G)$ であるとき，最適であるといわれる．グラフの真のパス幅は VLSI レイアウト，グラフの探索ゲーム [4]，グラフ埋め込み問題 [5, 6, 7] などに応用があることが知られている．

与えられた任意の N点グラフ G と整数 kに対して，$ppw(G) \leq k$であるか否かを判定する問題は NP 完全であるが，G が木に限られている場合には，最適な真のパス分解を $O(kN) = O(N \log N)$ 時間で求められることが知られている [4]．一方 kを定数に限った場合は，この問題は P に属すことが知られているが [2]，$k \geq 3$ の場合の多項式時間アルゴリズムは知られていない．$k = 1$ である場合は G がパスの部分グラフか否かを判定する問題と等価であるので，極めて簡単かつ効率的に解くことができる．$k = 2$ の場合，マイナー理論に基づく $ppw(G) \leq 2$ であるための必要十分条件が知られており，その特徴づけに基づいた $O(N \log^2 N)$ 時間アルゴリズムが知られている [2]．しかし，このアルゴリズムの計算時間には天文学的に大きな定数係数が含まれており，現実的な意味で効率的であるとはいえないし，真のパス分解を与えないという意味で構成的でもない．グラフレイアウトへの応用において，最大次数 3 以下で真のパス幅が 2 以下であるグラフの最適な真のパス分解を求めることは，レイアウトアルゴリズムの核となる手順になっている [5]．本稿では最大次数 3 以下のグラフ G の真のパス幅が 2 以下であるための必要十分条件を示し，これに基づいた G の幅 2 の真のパス分解を求める実際的な線形時間アルゴリズムを示す．

2 準備

集合Ωの要素または部分集合の系列 $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ に対して，$X_1(X_r)$ をそれぞれ $\mathcal{X}$の先頭 (終端) 要素という．系列 $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ の順序を反転させた系列 $(X_r, X_{r-1}, \ldots, X_1)$ を $\mathcal{X}^{-1}$ で表す．要素のない系列を nul で表す．2 個の系列 $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ と $\mathcal{Y} = (Y_1, Y_2, \ldots, Y_q)$ に対して，系列 $(X_1, X_2, \ldots, X_r, Y_1, Y_2, \ldots, Y_q)$ を $\mathcal{X} + \mathcal{Y}$で表す．集合$\Omega$の部分集合の系列 $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ と $W \subseteq \Omega$に対して，$(X_1 \cup W, X_2 \cup W, \ldots, X_r \cup W)$ を $\mathcal{X} \cup W$で表す．$(X_1 \cap W, X_2 \cap W, \ldots, X_r \cap W)$ を $\mathcal{X} \cap W$で表す．

グラフ G の点 vに接続する辺の集合を$\Gamma_G(v)$ で表す．$|\Gamma_G(v)|$ を vの次数と呼び，$\deg_G(v)$ で表す．$\Delta(G) = \max\{\deg_G(v) \mid v \in V(G)\}$ とする．$U \subseteq V(G)$ によって誘導される G の部分グラフを $G[U]$ で表す．$G[V(G) - U]$ を $G - U$で表す．$S \subseteq E(G)$ によって誘導される G の部分グラフを $G[S]$ で表す．グラフ G と Hに対して，点集合 $V(G) \cup V(H)$ および辺集合 $E(G) \cup E(H)$ からなるグラフを $G \cup H$で表す．根つき木 Tに対して，$v \in V(T)$ を根とする Tの部分木を $T(v)$ で表す．パスはそれ自身グラフであるけれども，我々はパス Pを，連続する 2 点が Pの辺によって結ばれているような点の系列によって表すことがある．

$E(G)$ が 2 個の部分集合 E_1，E_2に分割され，$G[E_1]$ と $G[E_2]$ がちょうど 1 点 vだけ共有するとき，vを G の切断点という．切断点をもたない連結グラフをブロックという．少なくとも 3 点をもつブロックはすべて 2-連結である．グラフ G の部分グラフ Hは，それ自身がブロックで，しかも Hを含むより大きなブロックがないとき，G のブロックといわれる．一つの外領域にすべての点が存在するような平面描画をもつグラフを外平面グラフと呼ぶ．外領域に含まれる辺を外辺，それ以外の辺を内辺と呼ぶ．外平面グラフ G の点の部分集合 $U = \{u_1, \ldots, u_l\}$ に対して，u_1と u_lが辺によって，u_iと u_{i+1} $(1 \leq i \leq l-1)$ が外辺によってそれぞれ結ばれ，かつ $U - \{u_1, u_l\}$ に属す全ての点が内辺に接続しないとき，$G[U]$ を終端領域という．任意の 2 連結外平面グラフ G は少なくとも 1 個の終端領域をもち，G が内辺をもつときは G は 2 個以上の終端領域をもつ．

3 グラフの幅 2 の真のパス分解を求めるアルゴリズム

グラフ G より自己閉路を取り除き，多重辺を 1 本の辺で置き換えて得られる単純グラフを G'とすると，真のパス分解の定義から，G の真のパス分解は G'の真のパス分解であり，その逆も成り立つ．したがって，G'の最適な真のパス分解は G のそれでもある．また，グラフ G の最適な真のパス分解は，G の各連結成分の最適な真のパス分解を連結させることにより得られる．これらの事実から，以下の議論では連結単純グラフのみを考える．

3.1 真のパス幅が 2 である最大次数 3 以下のグラフ

真のパス幅が 2 以下であるような木に対してはさまざまな特徴づけが知られているが [4, 7]，ここでは以下の特徴づけを用いる．

定理 A ([7]) 任意の木 Tおよび整数 $k \geq 2$ に対して，$ppw(T) \leq k$であるための必要十分条件は $ppw(T - V(P)) \leq k - 1$ であるようなパス Pが存在することである □

この定理の条件を満たすパスを Tの k-背骨と呼ぶ．

グラフ G に対して，$\Delta(G) \leq 3$ ならば G のトポロジカルバンド幅は G の混合探索数[1]に一致することが示されている [1]．[1] にはさらに G のトポロジカルバンド幅が 2 以下であるグラフの必要十分条件が示されている．一方 [7] には G の混合探索数が $ppw(G)$ に等しいことが示されている．以上の事実を組み合わせることにより，$\Delta(G) \leq 3$ のとき $ppw(G) \leq 2$ であるような G の必要十分条件が得られる．

補題 1 $\Delta(G) \leq 3$ なる 2-連結グラフ G に対して，$ppw(G) = 2$ であるための必要十分条件は G が外平面グラフであり，かつ 2 個以下の終端領域を持つことである． □

定理 2 G を $\Delta(G) \leq 3$ であるグラフとする．$ppw(G) \leq 2$ であるための必要十分条件は G に以下の条件を満たすブロックの系列 $\mathcal{B} = (B_1, B_2, \ldots, B_m)$ が存在することである．

1. $ppw(B_i) \leq 2$ $(1 \leq i \leq m)$.

2. $1 \leq i < m$ に対して，B_iと B_{i+1}が切断点 a_iをともに含み，a_0と a_mを B_1と B_mから条件 *3,4,5* を満たすように適当に選ぶことができる．

3. すべての $1 \leq i \leq m$ に対して，$a_{i-1} \neq a_i$.

4. もし B_i $(1 \leq i \leq m)$ が 2-連結成分で終端領域を 2 個もつならば，a_{i-1}と a_iは異なる終端領域に含まれる．

5. $G - \sum_{1 \leq i \leq m} B_i$の各連結成分は以下の条件を満たすようなパスに限る．

   (a) 各 a_i $(1 \leq i < m)$ に対して，高々1 本のパス Pが，a_iと Pの任意の 1 点を辺で結ぶことにより付加されている．

   (b) B_i $(1 \leq i < m)$ が 2-連結成分であるとき，各切断点 $a \in \{a_{i-1}, a_i\}$ の 1 個の隣接点 x に，高々1 本のパス Pが，x と Pの 1 端点を辺で結ぶことにより付加されている．

□

[1] [1] においては点探索数と呼ばれている．

3.2 アルゴリズム

N点からなる木 Tの最適な真のパス分解を求めるアルゴリズムは文献 [4] に述べられている．このアルゴリズムは $ppw(T)$ を $O(N)$ 時間で，さらに Tの最適な真のパス分解を $O(N ppw(T))$ 時間でそれぞれ求められるので，Tの幅 2 の真のパス分解は線形時間で求めることができる．ここでは$\Delta(G) \leq 3$ であるような閉路を含むグラフ G を考え，G の幅 2 の真のパス分解を求めるアルゴリズムを示す．図 1に示した PPD_GENERAL が所望の手続きであり，図 2，3に示した手続きはすべてサブルーチンである．

3.3 アルゴリズムの概要

G の切断点の集合を Uとする．G から Uを得る線形時間アルゴリズムが存在することはよく知られた事実である．1 辺からなるブロックにのみ含まれているような切断点の集合を U'とし，$A = U - U'$とする．A の各要素を関節点と呼ぶ．関節点 $p \in A$ は G の切断点でもあるから，$E(G)$ は A によって以下の条件を満たす互いに素な集合 $E_1, \ldots, E_m$に分割される:$i \neq j$に対して，$G[E_i]$ と $G[E_j]$ は高々1 個の関節点を共有する．$\mathcal{D} = \{G[E_i] \mid 1 \leq i \leq m\}$ とする．$D \in \mathcal{D}$は 2-連結であるとき，2-連結成分と呼ばれ，次数 3 の点を含む木であるとき，木成分と呼ばれ，次数 3 の点を含まない

```
Procedure PPD_GENERAL ( G )
[ Input:   a connected graph G with Δ(G) ≤ 3
  Output:  the proper-path-decomposition
           of G with width 2 ]
 1. let A be the set of articulation points of G;
 2. let H be the set of biconnected components of G;
 3. let T be the set of tree components of G;
 4. let P be the set of path components of G;
 5. if T = P = ∅ then
    (a) {a_0, a_m} :=END-REGION(the component in H);
    (b) goto 10
 6. find a sequence C = (C_1, ..., C_m) of components and
    {a_i | 1 ≤ i ≤ m − 1} s.t.
    (a) every H ∈ H ∪ T is contained in C,
    (b) for 1 ≤ i ≤ m − 1, C_i and C_{i+1} share {a_i}, and
    (c) if C_i ∈ H (1 ≤ i ≤ m) then for a ∈ {a_{i−1}, a_i},
        at most one path component not contained in
        C is attached to a vertex adjacent to a;
 7. if C does not exist then reject ;
 8. if C_1 ∈ H then a_0 := END-REGION1(C_1, a_1);
    else a_0 := 2-SPINE1(C_1, a_1);
 9. if C_m ∈ H then a_m := END-REGION1(C_m, a_m);
    else a_m := 2-SPINE1(C_m, a_m);
10. S = nul;
11. for i = 1 to m do
    (a) if C_i ∈ H then
        i. S := S + PPD_BCG2(C_i, a_{i−1}, a_i);
        ii. if there exists a path component P =
            (p_0, ..., p_l) s.t. (p_0, a_{i−1}) ∈ E(C_i) then
            S := S + PPD_PATH(P)^{−1} ∪ {a_{i−1}};
        iii. if there exists a path component P =
            (p_0, ..., p_l) s.t. (p_0, a_i) ∈ E(C_i) then S :=
            S + PPD_PATH(P) ∪ {a_i};
    (b) else S := S + PPD_TREE2(T, a_{i−1}, a_i);
12. return (S)
End
```

図 1: 一般のグラフの幅 2 の真のパス幅を求めるアルゴリズム

木であるとき，パス成分と呼ばれる．2-連結成分からなる集合を $\mathcal{H}$，木成分からなる集合を $\mathcal{T}$，パス成分からなる集合を $\mathcal{P}$とする．定理2より，$ppw(G) \leq 2$ であるための，次のような必要十分条件が得られることを示すのは容易である．

定理 3 $\Delta(G) \leq 3$であるグラフGに対して，$ppw(G) \leq 2$ であるための必要十分条件は G に以下の条件を満たす 2-連結成分，木成分，パス成分の系列 $\mathcal{C} = (C_1, C_2, \ldots, C_m)$ が存在することである．

1. $ppw(C_i) \leq 2$ $(1 \leq i \leq m)$.

2. $1 \leq i < m$ に対して，C_iと C_{i+1}が関節点 a_iをともに含み，a_0と a_mを C_1と C_mから条件 *3,4,5* を満たすように適当に選ぶことができる．

3. すべての $1 \leq i \leq m$ に対して，$a_{i-1} \neq a_i$.

4. もし C_i $(1 \leq i \leq m)$ が 2-連結成分で終端領域を 2 個もつならば，a_{i-1}と a_iは異なる終端領域に含まれる．

5. $G - \sum_{1 \leq i \leq m} C_i$の各連結成分は以下の条件を満たすようなパス成分に限る: C_i $(1 \leq i < m)$ が 2-連結成分であるとき，各関節点 $a \in \{a_{i-1}, a_i\}$ の 1 個の隣接点 x に，高々1 本のパス Pが，x と Pの 1 端点を辺で結ぶことにより付加されている．

□

$\Delta(G) \leq 3$ より，定理2の条件 5(a) を満たすパスは，2-連結成分に含まれる関節点には付加されず，木成分の一部となるので，定理3においてはこれに相当する条件が存在しないことに注意されたい．また，$ppw(G) \leq 2$ ならば，すべての 2-連結成分とパス成分は $\mathcal{C}$の要素でなくてはならないことにも注意されたい．

PPD_GENERAL の大まかな手順を以下に示す．

1. $\mathcal{H} \cup \mathcal{T}$の全要素を含み，定理3の条件 2,3,5 を満たすような連結成分の系列 $\mathcal{C}$を求める．
2. $\mathcal{C}$の各要素 C_i $(1 \leq i < m)$ に対して，$a_{i-1}(a_i)$ をそれぞれ真のパス分解の先頭 (終端) 要素に含むような，幅 2 の真のパス分解 $\mathcal{X}^i$を求める．
3. $\mathcal{C}$の各要素 C_i $(1 \leq i < m)$ が 2-連結成分であるとき，各関節点 $a_{i-1}(a_i)$ の 1 個の隣接点 x に，高々1 本のパス Pが，x と Pの 1 端点 p を辺で結ぶことにより付加されているとき，p が終端 (先頭) 要素に含まれるような Pの幅 1 の真のパス分解 $\mathcal{L}^i(\mathcal{R})^i$ を求める．
4. $\sum_{1 \leq i \leq l} [(\mathcal{L}^i \cup \{a_{i-1}\} + \mathcal{X}^i + \mathcal{R}^i \cup \{a_i\}]$ を出力して終了．

定理 3の条件 2,3,5 を満たすような連結成分の系列 $\mathcal{C}$ が存在するか否かを判定することと，そのような $\mathcal{C}$を実際に構成することは，$\mathcal{D}$の各要素と，関節点の関係を調べれば実現できる．$a_0(a_m)$ を決定するためには，$C_0(C_m)$ が 2-連結成分のときには，補題 1に基づいて，$C_0(C_m)$ より $a_1(a_{m-1})$ を含まない終端領域を探す手続き END-REGION1(図 2) を実行し，$C_0(C_m)$ が木また

はパス成分のときには，補題Aに基づいて，$a_0(a_m)$ と $a_1(a_{m-1})$ を端点にもつ2-背骨を探す手続き 2-SPINE1(図3)を実行する．

系列$\mathcal{C}$にしたがって，各連結成分C_iの所望の真のパス分解を求める手続きは，C_iが2-連結成分か木(パス)成分かによって PPD_BCG2(図2)と PPD_TREE2(図3)を使い分ける．どちらもa_{i-1}とa_iを，それぞれ先端要素と終端要素に含むような真のパス分解を与える手続きである．これら手続きの正当性は後に補題6，7に示される．

系列$\mathcal{C}$にしたがって，求めた真のパス分解と，a_i $(1 \le i \le l-1)$ の隣接点に付加されたパス成分の真のパス分解を上記のように混合して出力することにより，Gの幅2の真のパス分解が得られることを確かめるのは，困難ではない．

図1，2，3に示した手続きのうち，再帰呼び出しを行なうのは PPD_BCG2 と 2-SPINE1 であるが，どちらも次の再帰呼びだしでデータが少なくとも1点以上少なくなり，かつ1度のフェイズである1点の次数程度のステップ数しか要しないので，線形時間で終了する．他の手続きが線形時間で終了するのも明らかである．

以下の補題は補題6，7の証明のために用いられる．

補題 B ([3]) グラフ G に対する真のパス分解を $(X_1,\ldots,X_r)$ とすると，任意の $v \in V(G)$ は連続するX_iに含まれる． □

補題 4 連結グラフGに対する幅2の真のパス分解を $\mathcal{X} = (X_1,\ldots,X_r)$ とする．$s \in X_1$と$t \in X_r$を結ぶ任意のパスPに対して，$G - V(P)$ の連結成分はパスである．

証明 $\mathcal{X} \cap (V(G) - V(P))$ を $\mathcal{X}' = (X'_1,\ldots,X'_r)$ で表すと，明らかに$\mathcal{X}'$は$G - V(P)$に対して，真のパス幅の定義(ii)(iii)(iv)を満たす．$\mathcal{X}'$より冗長な要素を取り除いた系列が幅1の真のパス分解になっていることを示せば十分である．そのためには以下の(a)(b)が成り立つことを示せばよい: (a) 任意の$1 \le i \le r$に対して，$|X'_i| \le 2$である; (b) 任意の$1 \le a \le r$と$a+1 < c < r$に対して，$X'_a = X'_c$であるかまたは$|X'_a \cap X'_c| = 0$である．

$\mathcal{X}$は幅2であるから，真のパス幅の定義(iii)と補題Bより，すべての$1 \le i \le r$に対して，X_iは$V(P)$の要素を含む．したがって，$|X'_i| \le 2$であり，(a)は成り立つ．

真のパス分解の定義(v)より，任意のa,b,c $(1 \le a < b < c \le r)$ に対して，

$$|X_a \cap X_c| \le |X_b| - 2 = 3 - 2 = 1 \qquad (1)$$

である．a,b,c $(1 \le a < b < c \le r)$ に対して，$p_a \in X_a \cap V(P)$，$p_b \in X_b \cap V(P)$，$p_c \in X_c \cap V(P)$ とする．
場合 1 $p_a = p_c$のとき．(1)より，$|X_a \cap X_c| = 1$であるから$|X'_a \cap X'_c| = 0$が成り立つ．
場合 2 $p_a \ne p_c$のとき．$|X'_a \cap X'_c| = 1$を仮定し，このとき$X'_a = X'_c$であることを示す．$v \in X'_a \cap X'_b$で表すと，補題Bより，$v \in X_b$である．もし，$x \in X_b - (V(P) \cup \{v\})$ ならば，$|X_b| \le 3$より，$X_b \cap V(P) = \{p_b\}$である．Pは$s \in X_1$と$t \in X_r$の点を結んでいるから，$1 < b < r$より，$p_b \in X_{b-1} \cap X_{b+1}$である．また，補題Bより，$v \in X_{i-1} \cap X_{i+1}$なので，$|X_a \cap X_c| \ge |\{p_b, v\}| = 2$となり(1)に矛盾する．以上から，$\bigcup_{a<b<c} X_b$には$v$と$V(P)$以外の点は含まれない．したがって$X'_a = X_{a+1} = \cdots = X'_c$である．以上から(b)は成り立つ． □

補題 5 外平面成分 Bの任意の幅2の真のパス分解 $(X_1,\ldots,X_l)$ について，X_1, X_lには終端領域以外の点が含まれることはなく，さらに，Bが終端領域を2個もつならば，X_1とX_lにはそれぞれ異なる終端領域の点が含まれる．

証明 $s \in X_1$，$t \in X_l$とする．一般性を失わずsが終端領域に含まれない点であると仮定し，矛盾を導く．Bは3点以上の2連結成分であるから，sとtを結ぶ2本の点非共有パスが存在する．sは終端領域に含まれないので，それらのパスの少なくとも一方に対して，そのパス上の2点を結ぶ内辺が存在する．そのようなパスをP_1とし，他方をP_2とすると，BからP_2を除去したグラフには閉路が存在する．これは補題4に反する．

また，Bが終端領域が2個持っていて，かつsとtが同一の終端領域に含まれていると仮定しても，同様に矛盾を導くことができる． □

証明 補題4と2-背骨の定義から明らか． □

補題 6 $H \in \mathcal{H}$，$s,t \in V(H)$ $(\deg_H(s) = \deg_H(t) = 2)$に対して，手続き PPD_BCG2 は，$s \in X_1$かつ$t \in X_r$であるような$H$の幅2の真のパス分解 $\mathcal{X} = (X_1,\ldots,X_r)$ を，存在するときは必ず出力する．

証明 Hが外平面グラフで高々2個の終端領域をもつとき，s,tが異なる終端領域に含まれているならば，PPD_BCG2 は$s \in X_1$，$t \in X_r$，かつ$s \notin X_i$ $(2 \le i)$ となるようなHの幅2の真のパス分解を与えることをHの点数

に関する帰納法で示す．Hが内辺を持たないときは自明．よって，Hが内辺を持つと仮定する．したがって $|V(H)| > 3$ である．$|V(H)| = 4$ のときは明らか．$|V(H)| - 1$ なる H'と，その異なる終端領域に含まれる 2 点 s', t' ($\deg_H(s') = \deg_H(t') = 2$) に対して，成り立つと仮定する．$x, y$を $\deg_H(x) \leq \deg_H(y)$ なる s の隣接点とする．s, t は異なる終端領域に含まれるので，$\deg_H(x) < \deg_H(y)$ ならば明らかに $x \neq t$ である．したがって一般性を失わず，$x \neq t$ と仮定してよい．このとき s を yに縮約して得られるグラフ H'に対して，もし $\deg_H(x) = 2$ であるならば，明らかに $\deg_{H'}(x) = 2$ であり，かつ x は H'の終端領域に含まれる．また $\deg_H(x) = 3$ ならば，$\deg_H(y) = 3$ であり，このとき明らかに (x, y) は Hの内辺であるから，やはり $\deg_{H'}(x) = 2$ である．ゆえに帰納法の仮定から，PPD_BCG1 に H'と，x, t を再帰的に入力することによって $x \in X_1$かつ $x \notin X_i$ $(2 \leq i)$ であるような H'の真のパス分解 $\mathcal{X}$を得ることができる．X_i $(2 \leq i)$ に x は含まれることはないので，$|\{s, x, y\} \cap X_2| \leq 1$ であり，また，X_i $(1 \leq i)$ に対して，$|X_i| = 3$ であることに注意すると，$(\{s, x, y\}) + \mathcal{X}$は所望の Hの幅 2 の真のパス分解になっている．

逆に Hが外平面グラフでないか，または3個以上の終端領域をもつか，または $e = (s, t)$ $(\deg_H(s) \leq \deg_H(t))$ が内辺であるか，または e が終端領域に含まれていない場合は，補題 1，4，5によって Hが所望の真のパス分解を持たない． □

補題 7 *$T \in \mathcal{T} \cup \mathcal{P}$と $u_i \in V(T)$ with $\deg_T(u_i) = 1$ $(i = 1, 2)$ に対して，手続き* PPD_TREE2 *は，$u_1 \in X_1$ かつ $u_2 \in X_r$であるような Tの幅 2 の真のパス分解 $\mathcal{X} = (X_1, \ldots, X_r)$ を，存在するときは必ず出力する．*

証明 Tに対して $ppw(T)$ を求める [4] に示されたアルゴリズムは，根つき木 Tと任意の $v \in V(T)$ に対して，$ppw(T(v))$ を線形時間で求めることができる．ここではこのアルゴリズムを PPW_TREE と呼び，サブルーチンとして用いる．2-SPINE2 は指定された次数 1 の点 $u_1, u_2 \in V(T)$ に対して，それらの点を端点とするような 2-背骨を出力する手続きである．これは PPW_TREE および補題 Aを利用して，u_1と u_2を結ぶ一意的なパスが 2-背骨か否かを判定するだけで，それぞれ達成できる．PPD-SPINE は木 Tと 2-背骨 $P = (p_1, p_2, \ldots, p_l)$ を入力として Tの幅 2 の真のパス幅を出力する手続きである．Pは 2-背骨であるので，$T - V(P)$ は Tにおいて p_iと辺で結ばれているパス H_i $(1 < i < l)$ の集まりになっている．

パスの幅 1 の真のパス分解は，隣接する 2 点を要素に持つ集合の系列で構成できることは明らかである PPD_PATH．P，H_i $(1 < i < l)$ の幅 1 の真のパス分解を，それぞれ $\mathcal{W} = (W_1, \ldots, W_l)$，$\mathcal{Y}_i$ $(1 < i < l)$ とすると，Tの幅 2 の真のパス分解は $(W_1) + [\sum_{1<i<l} \mathcal{Y}_i \cup \{p_i\} + (W_i)] + (W_l)$ で得られることは簡単に確かめられる．

□

謝辞

日頃御指導頂く，本学梶谷洋司教授に感謝する．有益な御助言をいただいた本学高橋篤司助手に感謝する．本研究は東京工業大学の CAD21 研究体の研究課題の一部として行なわれたものである．

参考文献

[1] F. S. Makedon, C. H. Papadimitriou, and I. H. Sudborough. Topological bandwidth. *SIAM J. Alg. Disc. Meth.*, 6(3):418–444, 1985.

[2] A. Takahashi, S. Ueno, and Y. Kajitani. Minimal forbidden minors for the family of graphs with proper-path-width at most two. to appear in *IEICE Trans. Fundamentals*.

[3] A. Takahashi, S. Ueno, and Y. Kajitani. Minimal acyclic forbidden minors for the family of graphs with bounded path-width. *Discrete Math.*, 127:293–304, 1994.

[4] A. Takahashi, S. Ueno, and Y. Kajitani. Mixed searching and proper-path-width. *Theoretical Computer Science*, 137:253–268, 1995.

[5] 松林, 上野. グラフの格子への辺負荷最小埋め込みの計算複雑度について. 情報処理学会 第 51 回全国大会 講演論文集 (1), pp. 65–66, 1995.

[6] 田湯, 上野. 2 分木のハイパーキューブへの埋め込みについて. 信学技報 CAS92–42∼45, 92(236):77–84, 1992.

[7] 田湯, 上野. パス幅の制限された 2 分木のパスと格子への効率的な埋め込み. 情処研報 95–AL–45, 95(39):1–8, 1995.

Procedure PPD_BCG2 (H , s , t)

Input:	a biconnected graph H distinct vertices s and t in H s.t. $\deg_H(s) = \deg_H(t) = 2$
Output:	the proper-path-decomposition $(X_1, \ldots, X_r)$ of H with width 2 s.t. $s \in X_1$ and $t \in X_r$

1. **if** $|V(H)| = 3$ **then return** ($(V(H))$)
2. **let** x and y be vertices adjacent to s s.t. $\deg_H(x) \leq \deg_H(y)$ and if $\deg_H(x) = \deg_H(y) = 2$ then $x \neq t$;
3. **let** H' be the graph obtained from H by contracting (s, y);
4. **return** ($(\{s, x, y\}) +$ PPD_BCG2(H', x, t));

End

Procedure END-REGION (H)

Input:	a biconnected graph H
Output:	two vertices in distinct end-regions of H

1. $U := \{u \in V(H) \mid \deg_H(u) = 2\}$;
2. **if** $|U| = 0$ **then reject** ;
3. **if** $|U| = |V(H)|$ **then return** (an edge in H);
4. $R := \emptyset$;
5. **for each** connected component P of $H[U]$ **do** /* P is not a cycle but a path. */
   (a) **let** s and t be end-vertices of P; /* s and t are not necessarily distinct. */
   (b) **let** x and y be distinct vertices in $V(H) - U$ s.t. $(s, x) \in E(H)$ and $(t, y) \in E(H)$;
   (c) **if** $(x, y) \in E(H)$ **then** $R := R \cup \{s\}$;
   (d) **if** $|R| = 2$ **then return** (R);

   endfor
6. **reject** ;

End

Procedure END-REGION1 (H, v)

Input:	a biconnected graph H
Output:	a vertex in an end-region of H which does not contain v

1. $U := \{u \in V(H) \mid \deg_H(u) = 2\}$;
2. **if** $|U| = 0$ **then reject** ;
3. **if** $|U| = |V(H)|$ **then return** (an edge in H);
4. $R := \emptyset$;
5. **for each** connected component P of $H[U]$ which does not contain v **do** /* P is not a cycle but a path. */
   (a) **let** s and t be end-vertices of P; /* s and t are not necessarily distinct. */
   (b) **let** x and y be distinct vertices in $V(H) - U$ s.t. $(s, x) \in E(H)$ and $(t, y) \in E(H)$;
   (c) **if** $(x, y) \in E(H)$ **then** $R := R \cup \{s\}$;
   (d) **if** $|R| = 2$ **then return** (R);

   endfor
6. **reject** ;

End

図 2: 2-連結成分の幅 2 の真のパス幅を求めるアルゴリズム

Procedure PPW_TREE (T, u)

[Input: a tree T rooted at u
Output: $ppw(T_v)$ for each vertex v in T
(T_v is the subtree rooted at v.)]

This algorithm is shown in [4].

End

Procedure 2-SPINE1 (T, u)

[Input: a tree T rooted at u
$u \in V(T)$ with $\deg_T(u) = 1$
($ppw(T_v)$ for each $v \in V(T)$ is known)
Output: 2-spine P of T with end-vertex u]

1. **if** u has no children **then return** ((u));
2. **if** u has a child v with $ppw(T_v) \geq 3$ **then reject** ;
3. **if** u has two children v_i with $ppw(T_{v_i}) = 2$ $(i = 1, 2)$ **then reject** ;
4. Choose a child v of u s.t. $ppw(T_v) = \max\{ppw(T_{v_i}) \mid v_i$ is a child of $u\}$;
5. **return** ($(u) +$ 2-SPINE1(T_v, v));

End

Procedure 2-SPINE2 (T, u_1, u_2)

[Input: a tree T rooted at u_1
$u_i \in V(T)$ with $\deg_T(u_i) = 1$ $(i = 1, 2)$
($ppw(T_v)$ for each $v \in V(T)$ is known)
Output: 2-spine P of T with end-vertices u_1 and u_2]

1. **let** $P = (u_1, p_1, p_2, \ldots, u_2)$ be a unique path connecting u_1 and u_2;
2. **for each** p_i **do**
   (a) **for each** child v of p_i except a vertex in P **do**
      i. **if** $ppw(T_v) \geq 2$ **then reject** ;
   endfor ;
   endfor ;
3. **return** (P);

End

Procedure PPD_PATH (P)

[Input: a set of paths $P = (p_0, p_1, \ldots, p_l)$
$(E(P) \subseteq \{(p_{i-1}, p_i) \mid 1 \leq i \leq l\})$
Output: the proper-path-decomposition $(X_1, X_2, \ldots, X_l)$ of P with width 1 s.t.
$p_i \in X_i \cap X_{i+1}$ $(1 \leq i \leq l-1)$ and
$p_0 \in X_1$ and $p_l \in X_l$ $(1 \leq i \leq l-1)$]

1. **if** $l = 1$ **then return** ($(\{p_1\})$);
2. **for each** $1 \leq i \leq l$ **do**
   (a) $X_i = \{p_{i-1}, p_i\}$;
   endfor ;
3. **return** ($(X_1, X_2, \ldots, X_l)$);

End

Procedure PPD_SPINE (T, u_1, u_2)

[Input: a tree T
$u_1, u_2 \in V(T)$ s.t. the path connecting u_1 and u_2 is a 2-spine of T
Output: the proper-path-decomposition $(X_1, \ldots, X_r)$ of T with width at most 2
s.t. $u_1 \in X_1$ and $u_2 \in X_r$]

1. **let** P be a path $(p_0, p_1, \ldots, p_l)$ $(p_0 = u_1, p_l = u_2)$;
2. $\mathcal{X} :=$ PPD_PATH(P);
3. $\mathcal{S} := nul$;
4. **for each** p_i **do**
   (a) H_i be the subgraph of $T - V(P)$ which consists of connected components each of which has a vertex adjacent to p_i in T; **/* H_i has no vertex with degree more then 2. */**
   (b) $\mathcal{Y}_i :=$ PPD_PATH(H_i);
   (c) $\mathcal{S} := \mathcal{S} + (X_i) + \mathcal{Y}_i \cup \{p_i\}$;
   endfor ;
5. **return** ($\mathcal{S}$);

End

Procedure PPD_TREE2 (T, u_1, u_2)

[Input: a tree T
$u_i \in V(T)$ with $\deg_T(u_i) = 1$ $(i = 1, 2)$
Output: the proper-path-decomposition $(X_1, \ldots, X_r)$ of T with width at most 2
s.t. $u_1 \in X_1$ and $u_2 \in X_r$]

1. **call** PPW_TREE(T, u_1);
2. $P :=$ 2-SPINE2(T, u_1, u_2);
3. **return** (PPD_SPINE(T, u, v));

End

図 3: 木 (パス) 成分の幅 2 の真のパス幅を求めるアルゴリズム