

入選

3-連結グラフの3分割アルゴリズム†

鈴木 均†† 高橋 奈穂美†† 西関 隆夫††
宮野 浩††† 上野 修一†††

ここで、グラフの3分割問題とは、(1)無向単純グラフ G 、(2)互いに異なる G の3個の点 a_1, a_2, a_3 、(3) $n_1+n_2+n_3=n$ なる自然数 n_1, n_2, n_3 を入力したときに、 G の三つの素な連結部分グラフ G_1, G_2, G_3 で各 G_i が a_i を含み、その点数が n_i であるものを求める問題である。 n はグラフ G の点数である。グラフ G が3-連結ならば3分割問題には必ず解が存在することを Györi と Lovász は独立に証明している。しかしその証明からは多項式時間アルゴリズムは得られない。なお、入力グラフ G が3-連結と限らない一般の場合には、3分割問題は NP-困難であることが知られている。本文では3-連結グラフ G の3分割問題を解く多項式時間のアルゴリズムを与える。そのアルゴリズムは二つのステップからなる。まず、3-連結性を保ったまま G から何本かの辺を除去して辺数を $O(n)$ に減らす。この部分は G の辺数に比例した時間で終了する。次に得られた辺数なグラフを用いて3分割問題を解く。この部分はグラフの辺数と点数の積に比例した時間で終了する。結局アルゴリズムは全体で $O(n^3)$ 時間で終了する。

1. ま え が き

$G=(V, E)$ を点集合 V 、辺集合 E からなる無向単純グラフとする。なお $n=|V|$ 、 $m=|E|$ とし、 $V=V(G)$ と書くことがある。また $V' \subset V$ によって誘導される G の部分グラフを $G[V']$ と書く。本文ではグラフ G が3-連結であるときに、グラフ3分割問題を $O(n^3)$ 時間で解くアルゴリズムを与える。グラフの3分割の例を図-1に示す。一般にグラフ k 分割問題とは以下のような(入力)から、(出力)を求める問題である。

(入力)

$G=(V, E)$: 無向単純グラフ
 a_1, a_2, \dots, a_k : 互いに異なる G の k 個の点
 n_1, n_2, \dots, n_k : $\sum_{i=1}^k n_i = n = |V|$ なる自然数

(出力)

(V_1, V_2, \dots, V_k) : 各 $i(1 \leq i \leq k)$ について次の(1)~(3)を満足する V の分割
(1) $a_i \in V_i$;
(2) $G[V_i]$ は連結である ;
(3) $|V_i| = n_i$.

グラフ G が k -連結ならば k 分割問題に必ず解が存在

することを Györi⁵⁾ と Lovász⁶⁾ は独立に証明している。特に $k=2$ の場合に対しては Györi の証明から多項式時間アルゴリズムがただちに得られる。しかし $k \geq 3$ の場合に対してはそれらの証明からは多項式時間アルゴリズムは得られない。本文では、 $k=3$ の場合に対し $O(n^3)$ 時間のアルゴリズムを与える。なお、 G が k -連結グラフとは限らない場合には、 G が二部グラフでかつ $n_i = |V|/k$ であると限定してもグラフ k 分割問題は NP-困難であることが知られている³⁾。なおグラフ分割問題は耐障害ルーティングなどに現れる⁶⁾。

2. 準 備

$G=(V, E)$ から $V' \subset V$ の全ての点を除去して得られるグラフを $G-V'$ と書く。特に V' が1点 v からなるとき、 $G-v$ と書くことがある。 G が連結なのに、 $G-v$ が非連結であるとき、点 v は切断点と呼ばれる。点 $v, w \in V$ に対し、 $G+(v, w)$ は G に辺 (v, w) を付加して得られる単純グラフとする。したがって $(v, w) \in E$ のときには $G+(v, w)$ は G そのものを表す。グラフ G が k 点からなる完全グラフ K_k であるか、あるいは $k+1$ 個以上の点をもちどの $k-1$ 点を取り除いても非連結にならないとき、 G は k -連結であるということにする。 G の辺 (v, w) の両端点を同一視し、それによって生じた自己ループや多重辺を取り除く操作を辺 (v, w) の縮約という。 G の辺 (v, w) を縮約して得られるグラフを $G/(v, w)$ と書く。 k -連結グラフ G について、 $G/(v, w)$ も k -連結グラフであるとき、辺 (v, w) は k -縮約可能であると

† An Algorithm for Tripartitioning 3-Connected Graphs by HITOSHI SUZUKI, NAOMI TAKAHASHI, TAKAO NISHIZEKI (Tohoku University, Department of Electrical Communications, Faculty of Engineering), HIROSHI MIYANO and SHUICHI UENO (Tokyo Institute of Technology, Department of Electrical and Electronic Engineering).

†† 東北大学工学部通信工学科

††† 東京工業大学工学部電気電子工学科

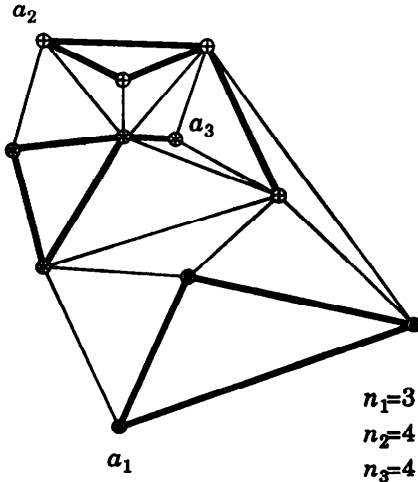


図-1 3-連結グラフ G とその3分割 (太線は $G[V_i]$)
 Fig. 1 3-connected graph G and its tripartition (graphs $G[V_i]$ are drawn in thick lines).

$n_1=3$ ● : vertex in V_1
 $n_2=4$ ⊕ : vertex in V_2
 $n_3=4$ ⊙ : vertex in V_3

いう。一方、 $G(v, w)$ が k -連結グラフではないとき、辺 (v, w) は k -縮約不可能であるという。次の定理が成立することが知られている⁴⁾。

【定理1】 $n \geq 3$ なる 2-連結グラフ G の任意の点には 2-縮約可能な辺が 2 本以上接続している。 □

3. では 2 分割問題を少し拡張した問題を解くアルゴリズムを与え、4. ではそのアルゴリズムを用いて 3 分割問題を解くアルゴリズムを与える。それらのアルゴリズムの計算時間は $O(mn)$ である。5. では任意の 3-連結グラフに対し、辺数が $O(n)$ 本の 3-連結全域部分グラフを求める線形時間アルゴリズムを与える。そのアルゴリズムを用いて与えられたグラフ G を 3-連結のまま辺数を $O(n)$ 本に減らし、得られた辺疎なグラフに 4. で与える 3 分割アルゴリズムを適用すれば、 $O(n^2)$ 時間でグラフ G を 3 分割できる。

3. 2-連結グラフの2分割アルゴリズム*

本章では 2-連結グラフの 2 分割問題を少し拡張した問題を解くアルゴリズム PART 2* を示す。この手続きは次章で示す 3 分割アルゴリズムで用いられる。

手続き PART 2* の入力と出力は以下のとおりである。

(入力)

$G=(V, E)$: 2-連結グラフ

a_1, a_2 : G の異なる 2 個の点

n_1, n_2 : $n_1+n_2 \geq n$ なる自然数
 u, w : G の異なる 2 個の点。ただし $\{a_1, a_2\} \cap \{u, w\} \neq \emptyset$ であってもよい。

(出力)

(V_1, V_2) : 以下の (1)~(3)'' を満足する V の分割

(1) $a_1 \in V_1, a_2 \in V_2$

(2) $G[V_1]$ と $G[V_2]$ が連結

(3)' $|V_1|=n_1$ または $(|V_1| < n_1$ かつ $V_1 \cap \{u, w\} \neq \emptyset$)

(3)'' $|V_2|=n_2$ または $(|V_2| < n_2$ かつ $V_2 \cap \{u, w\} \neq \emptyset$)

必ずしも $n_1+n_2=n$ とは限らないことに注意しよう。以下にアルゴリズム PART 2* を示す。

function PART 2* ($G, a_1, a_2, n_1, n_2, u, w$);

begin

if $|V(G)|=2$ then return ($\{a_1\}, \{a_2\}$)

else if $n_1=1$ then return ($\{a_1\}, V(G)-\{a_1\}$)

else if $n_2=1$ then return ($V(G)-\{a_2\}, \{a_2\}$)

else

begin

一般性を失うことなく $\{a_1, a_2\} \neq \{u, w\}$ ならば $a_1 \neq u, w$ であると仮定する; {必要なならば a_1 と a_2 の役割を交換する}

2-縮約可能辺 $(a_1, a) \neq (a_1, a_2)$ を見つける;

{定理 1 よりそのような辺は必ず存在する}

$G := G(a_1, a)$;

a_1 と a を同一視して得られる点を新たに a_1 とする;

$(V_1, V_2) := \text{PART 2}^*(G, a_1, a_2,$

$n_1-1, n_2, u, w)$;

return ($V_1 \cup \{a\}, V_2$)

end

end;

手続き PART 2* が上の (1)~(3)'' を満足する V の分割 (V_1, V_2) を求めることは点数 n の帰納法で容易に証明できる。

PART 2* の計算時間について示そう。2-縮約可能辺 $(a_1, a) \neq (a_1, a_2)$ を見つけるには、 a_1 に隣接する

* なお最近九州大学宮野悟助教授より、2-連結グラフの 2 分割問題は st -numbering を用いて線形時間で解けることが指摘された。

a_2 以外の点で $G - \{a_1\}$ の切断点ではないものを見つけられればよい。グラフの全ての切断点は深さ優先探索を1回行うことで見つけれられる。したがって、2-縮約可能な辺 (a_1, a) は $O(m)$ 時間で見つけれられる。また辺 (a_1, a) の縮約は明らかに $O(n)$ 時間で行える。したがって、PART 2* を再帰的に呼び出す部分以外は $O(m)$ 時間で実行される。再帰呼び出しは高々 n 回しか行われないので、PART 2* 全体の実行時間は $O(mn)$ である。

$n_1 + n_2 = n$ なる2分割問題を解くには互いに異なる任意の2点を u, w として選んで PART 2* を実行すればよい。このようにして2分割問題を解くアルゴリズムを PART 2 と書くことにする。

4. 3-連結グラフの3分割アルゴリズム

本章では3-連結グラフ G に対し3分割問題を解くアルゴリズム PART 3 を与える。

function PART 3 ($G, a_1, a_2, a_3, n_1, n_2, n_3$);

begin

(0) $(a_1, a_2), (a_2, a_3), (a_3, a_1)$ の中で、 E に含まれないものがあれば E に加える。こうしても不都合は生じない。なぜなら $G[V_i]$ ($i=1, 2, 3$) にこれらの辺が含まれることはないからである。

(1) n_1, n_2, n_3 のいずれかが1ならば、PART 2 を適用する。たとえば $n_1=1$ ならば、

$(V_2, V_3) := \text{PART } 2(G - a_1, a_2, a_3, n_2, n_3)$;

return $(\{a_1\}, V_2, V_3)$

を実行し、終了する。 $n_2=1$ または $n_3=1$ のときも同様である。

(2) a_1 に隣接し a_2 や a_3 ではない任意の点 v_a を選ぶ。辺 (a_1, v_a) が3-縮約可能ならば(3)へ、そうでないときは(4)へ行く。

(3) いま (a_1, v_a) は3-縮約可能である。 (a_1, v_a) を縮約して得られたグラフに PART 3 を適用する。すなわち、

$(V_1, V_2, V_3) := \text{PART } 3(G/(a_1, v_a), a_1, a_2, a_3, n_1 - 1, n_2, n_3)$;

return $(V_1 \cup \{v_a\}, V_2, V_3)$

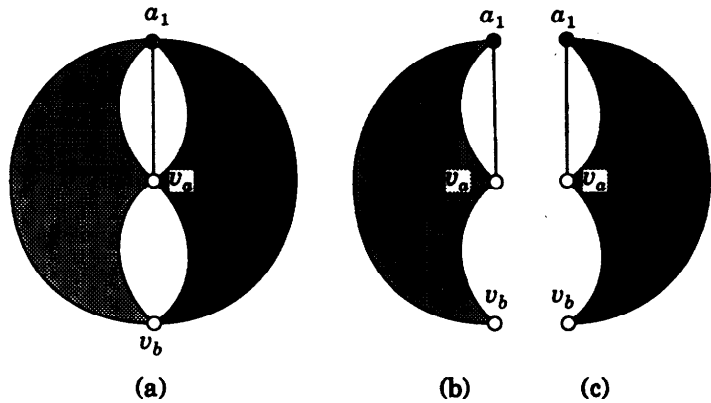


図-2 グラフ(a) G , (b) H , (c) J
Fig. 2 Graphs (a) G , (b) H and (c) J .

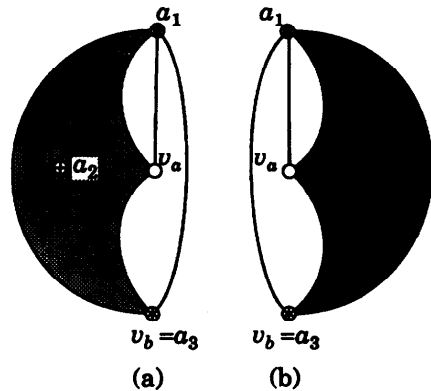


図-3 $v_b = a_3$ の場合のグラフ(a) H と (b) J
Fig. 3 Graphs (a) H and (b) J with $v_b = a_3$.

を実行し、終了する。

(4) いま (a_1, v_a) は3-縮約不可能である。したがって、 $G' = G - \{a_1, v_a, v_b\}$ が非連結になる点 $v_b \in V(G) - \{a_1, v_a, v_b\}$ が存在するので、そのような点 v_b を見つける。ただし、もし $G - \{a_1, v_a, a_2\}$ が非連結ならば a_2 を v_b として選び、もし $G - \{a_1, v_a, a_3\}$ が非連結ならば a_3 を v_b として選ぶことにする。 G' の連結成分で a_2 または a_3 を含むものの点集合を X とし、 $X \cup \{a_1, v_a, v_b\}$ から誘導される G の部分グラフを H とし、 $V - X$ から誘導される G の部分グラフを J とする(図-2 参照)。辺 (a_2, a_3) が存在するので $\{a_1, a_2, a_3\} \subset V(H)$ であることに注意されたい。 $v_b = a_2$ または a_3 ならば(5)へ、そうでなければ(6)へ行く。

(5) 一般性を失うことなく $v_b = a_3$ とする(図-3 参

照). 次の三つの場合(5 a), (5 b), (5 c)がある.

- (5 a) $V(H) - \{a_1, a_3\} \leq n_2$ のとき. この場合には $V(H) - \{a_1, a_3\}$ の全ての点を V_2 に含め, J に辺 (v_a, a_3) を加えた 3-連結グラフ $J+(v_a, a_3)$ に PART 3 を適用する. すなわち

(V_1, V_2, V_3) :
 $= \text{PART 3 } (J+(v_a, a_3), a_1, v_a, a_3,$
 $n_1, n_2 - |V_H| + 2, n_3);$

return $(V_1, V_2 \cup (V(H) - \{a_1, a_3\}), V_3)$
 を実行して終了する. なお $J+(v_a, a_3)$ は G から $V(H) - \{a_1, a_3\}$ の全ての点を v_a に同一視して得られるグラフである.

- (5 b) $|V(H) - \{a_1, a_3\}| > n_2$ かつ $|V(H - a_3)| \leq n_1 + n_2$ のとき. この場合には $H - a_3$ の全ての点を V_1 と V_2 に含める. すなわち

$(V_1, V_2) := \text{PART 2 } (H - a_3, a_1, a_2,$
 $|V(H - a_3)| - n_2, n_2);$

を実行する. もし $v_a \in V_1$ になれば
 $(V'_1, V_3) := \text{PART 2 } (J/(a_1, v_a), a_1, a_3,$
 $|V(J/(a_1, v_a))| - n_3, n_3);$

return $(V_1 \cup V'_1, V_2, V_3)$
 を実行する. もし $v_a \in V_2$ ならば
 $(V'_1, V_3) := \text{PART 2 } (J - v_a, a_1, a_3,$
 $|V(J - v_a)| - n_3, n_3);$

return $(V_1 \cup V'_1, V_2, V_3)$
 を実行する. なお $H+(v_a, a_3)$ および $J+(v_a, a_3)$ が 3-連結なので, $H - a_3, J/(a_1, v_a)$ および $J - v_a$ は 2-連結であることに注意されたい.

- (5 c) (5 a)でも(5 b)でもないとき. いま $|V(H - a_3)| > n_1 + n_2$ である. この場合は J の a_1, v_a 以外の点全てを V_3 に含める. すなわち

(V_1, V_2, V_3) :
 $= \text{PART 3 } (H+(v_a, a_3), a_1, a_2, a_3,$
 $n_1, n_2, |V(H)| - n_1 - n_2);$

return $(V_1, V_2, V_3 \cup (V(J) - \{a_1, v_a\}))$
 を実行する. $H+(v_a, a_3)$ は G から $V(J) - \{a_1, v_a\}$ の全ての点を a_3 に同一視して得られるグラフであり, 3-連結であることに注意しよう.

- (6) いま $a_2, a_3 \neq v_a$ であり, a_2 と a_3 は隣接して

いるので, ともに H に含まれる. $H' = H + (a_1, v_a)$ が 3-連結のときは(7)へ, そうでなければ(8)へ行く.

- (7) いま H' は 3-連結である.

- (7.a) $|V(H - a_1)| \geq n_2 + n_3$ ならば,

(V_1, V_2, V_3) :
 $= \text{PART 3 } (H', a_1, a_2, a_3,$
 $|V(H')| - n_2 - n_3, n_2, n_3);$
return $(V_1 \cup (V(J) - \{v_a, v_b\}), V_2, V_3)$

を実行し終了する.

- (7 b) $|V(H - a_1)| < n_2 + n_3$ ならば

$(V_2, V_3) := \text{PART 2}^*(H - a_1, a_2, a_3,$
 $n_2, n_3, v_a, v_b)$

を実行する. 次に 3-連結グラフ $J' = J + (a_1, v_a) + (v_a, v_b)$ を作る. $v_a \in V_2, v_b \in V_3$ あるいは $v_a \in V_3, v_b \in V_2$ ならば(7 b I)へ, $v_a, v_b \in V_2$ あるいは $v_a, v_b \in V_3$ ならば(7 b II)へ行く.

- (7 b I) 一般性を失うことなく $v_a \in V_2, v_b \in V_3$ とする.

(V_1, V'_2, V'_3) :
 $= \text{PART 3 } (J', a_1, v_a, v_b,$
 $n_1, n_2 - |V_2| + 1, n_3 - |V_3| + 1);$
return $(V_1, V_2 \cup V'_2, V_3 \cup V'_3)$

を実行し終了する.

- (7 b II) 一般性を失うことなく $v_a, v_b \in V_2$ とする. この場合は $V_3 \cap \{v_a, v_b\} = \emptyset$ なので, 条件(3)*により $|V_3| = n_3$ のはずである. したがって

$J'' := J' / (v_a, v_b);$
 $(V_1, V'_2) := \text{PART 2 } (J'', a_1, v_a,$
 $n_1, |V(J'')| - n_1);$

return $(V_1, V_2 \cup V'_2, V_3)$

を実行し終了する. なお J'' は 2-連結である.

- (8) いま H' は 3-連結ではないが図-4(a)のような 2-連結グラフである. 後で補題 1 で示すように $H'' = H' - a_1$ は図-4(b)のような 2-連結成分の“鎖”になっている. H'' の 2-連結成分のうち a_2, a_3 を含むものを F とする. H'' の切断点と v_a, v_b のうちで F に含まれるものを u, w とする. F の点数を n_F とする. $n_F > n_2 + n_3$ ならば(9)へ, $n_F \leq n_2 + n_3$ ならば(10)へ行く.

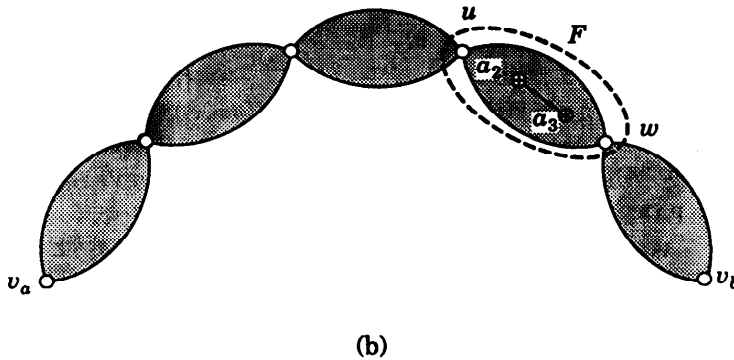
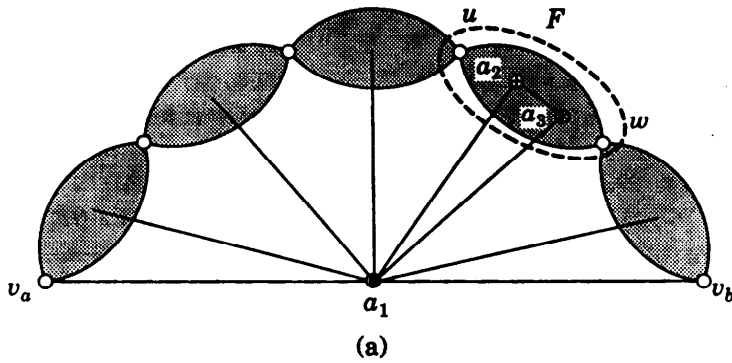


図-4 2-連結グラフ(a) H' と連結グラフ(b) H''
 Fig. 4 (a) 2-connected graph H' and (b) H'' .

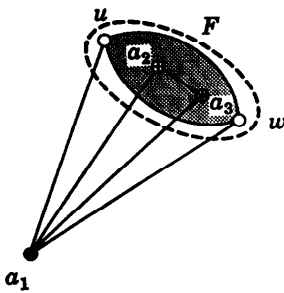


図-5 3-連結グラフ M
 Fig. 5 3-connected graph M .

(9) いま $n_F > n_2 + n_3$ である。 $V_1 := V - V(F)$ とする。 $V(F) \cup \{a_1\}$ から誘導される H' の部分グラフに辺 (a_1, u) , (a_1, w) を加えたグラフ M を作り (図-5 参照)。

$(V_1, V_2, V_3) := \text{PART 3}(M, a_1, a_2, a_3, n_F - n_2 - n_3 + 1, n_2, n_3);$

return $(V_1 \cup V_1, V_2, V_3)$

を実行し終了する。なお、明らかに M は 3-連

結である。

(10) いま $n_F \leq n_2 + n_3$ である。また $n_F \geq 3$ である。なぜならもし $n_F = 2$, すなわち F が 1本の辺からなるとすると, $\{u, w\} = \{a_2, a_3\}$ であり, $G - \{a_1, v_a, a_2\}$ あるいは $G - \{a_1, v_a, a_3\}$ が非連結になり, (4)の v_b の選び方に矛盾してしまうからである。

$(V_2, V_3) := \text{PART 2}^*(F, a_2, a_3, n_2, n_3, u, w)$ を実行する。 $u \in V_2, w \in V_3$ または $u \in V_3, w \in V_2$ の場合は (11)へ, $u, w \in V_2$ または $u, w \in V_3$ の場合は (12)へ行く。

(11) 一般性を失うことなく $u \in V_2, w \in V_3$ とする。図-6 のように $G - (V(F) - \{u, w\})$ に辺 (a_1, u) , (a_1, w) , (u, w) を加えたグラフを B とする。

$(V_1, V_2, V_3) := \text{PART 3}(B, a_1, u, w, n_1, n_2 - |V_2| + 1, n_3 - |V_3| + 1);$

return $(V_1, V_2 \cup V_2, V_3 \cup V_3)$

を実行し終了する。なお B は 3-連結である。

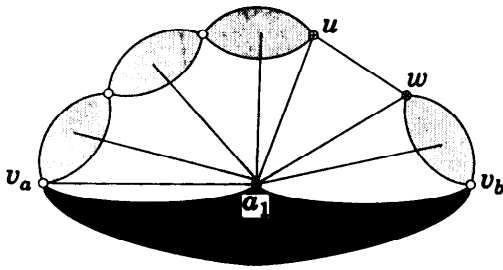


図-6 3-連結グラフ B
Fig. 6 3-connected graph B.

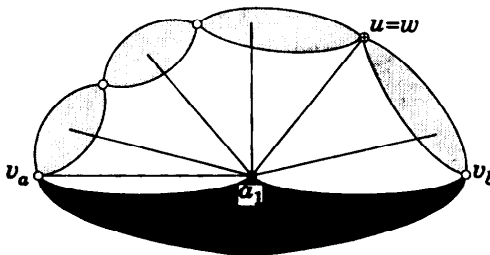


図-7 2-連結グラフ B'
Fig. 7 2-connected graph B'.

また $n_F \geq 3$ であるから $|V(B)| < |V(G)|$ である。

- (12) 一般性を失うことなく $u, w \in V_2$ とする。このとき $V_3 \cap \{u, w\} = \emptyset$ なので条件(3)"により $|V_3| = n_3$ である。図-7のように G において F の全ての辺を1点に縮約したグラフを B' として、

```
(V1, V2) := PART 2 (B', a1, u,
                    n1, |V(B')| - n1);
return (V1, V2 U V2, V3)
```

を実行し終了する。 B' は 2-連結であることに注意されたい。

end (of function PART 3);

次の補題を用いればアルゴリズムの正当性は点数の帰納法により容易に証明できるので、その証明は省略する。

【補題 1】 アルゴリズム PART 3 の(8)で H'' は連結であるが、2-連結ではない。 u_i が H'' の切断点であるとき、 $H'' - u_i$ には連結成分がちょうど二つあり、それぞれの成分は v_a あるいは v_b を含む。

(証明) H' は 2-連結だから、 H'' は連結である。 H' は 3-連結ではないので、切断点对 $[v_1, v_2]$ が存

在する。 $H' + (v_a, v_b)$ は 3-連結だから、 $H' - \{v_1, v_2\}$ に連結成分はちょうど2個あり、各連結成分は v_a あるいは v_b を含む。一方 $(a_1, v_a), (a_1, v_b)$ なる辺が存在するので、 a_1 はいずれの成分にも属し得ない。よって、 $a_1 \in \{v_1, v_2\}$ である。したがって、 $H'' = H' - a_1$ には切断点が存在し、2-連結ではない。もちろん H'' の切断点のひとつを v とすると、 $H'' - v = H' - \{v, a_1\}$ には成分がちょうど二つあり、それぞれ v_a または v_b を含む。(証明終)

補題 1 より、図-4(b)のように H'' が 2-連結成分の鎖になっていることが分かる。

PART 3 の計算時間について考察しよう。(1)で終了する場合は明らかに計算時間は $O(mn)$ である。

(2)で辺 (a_1, v_a) が 3-縮約可能かどうかを調べるには $G - \{a_1, v_a\}$ に切断点があるかどうかを調べればよい。これは深さ優先探索を1回行うことで調べることができる。したがって(2)は $O(m)$ 時間で実行できる。また(4)で v_b を見つける部分は(2)と同時に行える。

(4)のそれ以外の部分は明らかに $O(m)$ 時間で実行できる。(6)で H' が 3-連結かどうかを調べるには $H' - a_1$ が切断点をもつかどうかを調べればよい。したがって(6)は $O(m)$ 時間で実行できる。また(6)で、 $H' - a_1$ の全ての切断点を見つけていることができるので、(8)で F を見つけるのに要する時間は明らかに $O(m)$ であることが分かる。以上より PART 3 を再帰呼び出しする部分と、PART 2, PART 2* を用いる部分以外に要する時間は $O(m)$ であることが分かる。アルゴリズムは(1)以外では(3), (5), (7), (9), (11), (12)で終了する。PART 2, PART 2*, PART 3 の適用のしかたには以下の3通りがある。

- (A) PART 2 と PART 2* だけを適用する。
(5 b), (7 b II), (12)
- (B) PART 3 だけを G より点数が少ないグラフに適用する。
(3), (5 a), (5 c), (7 a), (9)
- (C) PART 2* を適用し、その結果どの V_i に含まれるかが決定した点を除去したグラフに PART 3 を適用する。
(7 b I), (11)

PART 3 は必ず点数が少なくなったグラフに対し再帰的に呼び出されることに注意されたい。したがって、PART 3 が再帰呼び出しされる回数は高々 n 回である。よってアルゴリズム中で付加される辺の本数は $O(n)$ 本である。以上により、アルゴリズム PART 3 が $O(mn)$ 時間で終了することが分かる。

5. 3-連結全域部分グラフを見つける
アルゴリズム*

本章では、任意の 3-連結グラフ $G=(V, E)$ が与えられたときに、辺数が $O(n)$ 本の 3-連結全域部分グラフ G_3 を求める線形時間アルゴリズムを与える。

一般に点数 $n \geq 3k-2$ の k -連結グラフの任意の極小 k -連結全域部分グラフに辺は $kn/2$ 本以上あり、高々 $k(n-k)$ 本しかないことが知られている²⁾。また明らかに極小 k -連結全域部分グラフは多項式時間で求まる。本章で与えるアルゴリズムは辺数が $3n-5$ 本以下の 3-連結全域部分グラフを $O(m)$ 時間で求める。

$G_1=(V, E_1)$ を G の深さ優先探索木とする。 G_1 は G の根付き全域木である。 G_1 の各点 v の親、子、先祖、子孫を通常の意味で定義する。なお本文では v の先祖や子孫には v 自身を含まないものとする。 v の親を $p(v)$ と書く。 v の先祖、子孫の集合をそれぞれ $ANC(v)$, $DES(v)$ と書く。木あるいは深さ優先探索木に関する用語や性質については文献 1) を参照されたい。

$G_1=(V, E_1)$ は木であるから、(1-)連結である。また G_1 は深さ優先探索木であるから、 $(v, w) \in E$ ならば w は v の先祖か子孫である。 $dfnum(v)$ を点 $v \in V$ の深さ優先番号 (depth-first number) とする。 G において v に隣接する v の先祖で G_1 の根に最も近い点 ($dfnum$ が最小の点) を $top(v)$ と書く。そのような点が存在しない場合は $top(v)$ を定義しない。辺 $(v, top(v))$ の集合を E_2 と書く。すなわち

$$E_2 = \{(v, top(v)) \mid v \in V \text{ かつ } top(v) \text{ が定義される}\}.$$

【補題 2】 $G=(V, E)$ が 2-連結ならば、 $G_2=(V, E_1 \cup E_2)$ も 2-連結である。

(証明) 明らかに G_2 は連結である。ある点 $v \in V$ に対し $G_2 - \{v\}$ が連結でないと仮定する。 G は 2-連結だから $G - \{v\}$ に $DES(v)$ と $ANC(v)$ を結ぶ辺が存在する。したがって E_2 の選び方より $G_2 - \{v\}$ にも $DES(v)$ と $ANC(v)$ を結ぶ辺が存在するはずである。よって $G_2 - \{v\}$ は連結であり、仮定に矛盾する。(証明終)

各点 $v \in V$ について $ANC(p(v))$ と $DES(v) \cup \{v\}$ (図-8 参照) を結ぶ辺 (u, w) で $dfnum(u)$ が最大な

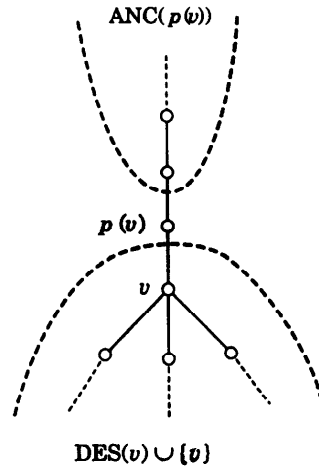


図-8 深さ優先探索木の一部
Fig. 8 Depth-first search tree.

ものの任意の 1 本を $e(v)$ とする (ただし $u \in ANC(p(v))$, $w \in DES(v) \cup \{v\}$ とする)。そのような辺がなければ $e(v)$ を定義しない。 $e(v)$ の集合を E_3 と書く。すなわち

$$E_3 = \{e(v) \mid v \in V \text{ かつ } e(v) \text{ が定義される}\}.$$

【補題 3】 G が 3-連結ならば、 $G_3=(V, E_1 \cup E_2 \cup E_3)$ も 3-連結である。

(証明) G_2 が 2-連結だから、 G_3 も 2-連結である。点 $v_1, v_2 \in V$ に対し $G_3 - \{v_1, v_2\}$ が連結でないと仮定しよう。一般性を失うことなく $dfnum(v_1) < dfnum(v_2)$ としてよい。もし v_1 が v_2 の先祖でなければ、 v_1 または v_2 は G_3 の切断点であることになり G_3 の 2-連結性に矛盾する。したがって v_1 は v_2 の先祖であるとしてよい。 v_1 の子で v_2 の先祖であるものを v とする。 $V - \{v_1, v_2\}$ を以下のように三つの点集合 X_A, X_B, X_C に分ける (図-9 参照)。

$$\begin{aligned} X_C &= DES(v_2) \\ X_B &= \{v_1\} \cup DES(v_1) - (X_C \cup \{v_2\}) \\ X_A &= V - (X_B \cup X_C \cup \{v_1, v_2\}) \end{aligned}$$

なお、 v_1 が存在しないとき、すなわち $p(v_2) = v_1$ のときは $X_B = \emptyset$ であるとする。以下では $X_A, X_B, X_C \neq \emptyset$ とするが、いずれかが空集合の場合も同様に証明できる。点集合 X_A, X_B, X_C から誘導される G_2 の部分グラフをそれぞれ A, B, C とする。 G_2 の 2-連結性と E_2 の選び方により A は連結である。また B も明らかに連結である。 C の連結成分を C_1, C_2, \dots, C_i とする。各 C_i の点集合は v_2 のひとつの子とその子孫からなる。 C_i の点が G_3 上で A, B の点に隣接す

* なお、これとは別の手法で、 k -(辺)連結グラフの k -(辺)連結全域部分グラフで、辺の本数が $k(n-(k+1))$ 以下のものを見つける $O(m)$ 時間のアルゴリズムが報告されている³⁾。

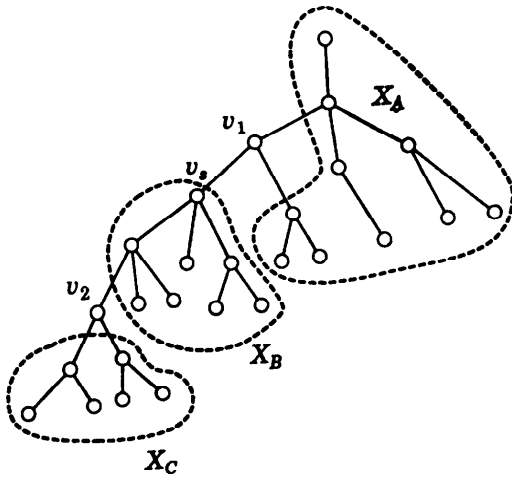


図-9 補題3の証明の説明図

Fig. 9 Illustration for the proof of Lemma 3.

るか否かで、 C_i を次の4通りに分類する。

A型: C_i の点で A の点に隣接するものがあり、 B の点に隣接するものはない。

B型: B に隣接する点があり、 A に隣接する点はない。

AB型: A に隣接する点と B に隣接する点の両方がある (同一の点でもよい)。

O型: A に隣接する点も B に隣接する点もない。

まずO型の C_i が存在しないことを示そう。 G は3-連結だから G には C_i と A を結ぶ辺か、 C_i と B を結ぶ辺が存在する。 C_i と A を結ぶ辺があれば、 E_2 の選び方よりそのうち少なくとも1本の辺は E_2 に含まれるはずであり、 C_i はA型もしくはAB型である。また C_i と B を結ぶ辺があれば、 v_2 の子である C_i の点 v_c について $e(v_c)$ が定義される。このとき $e(v_c) = (b, c)$ 、 $dfnum(b) < dfnum(c)$ 、とすると、 $b \in X_B$ 、 $c \in V(C_i)$ であるから、 C_i はB型もしくはAB型のはずである。したがって、 C_i はO型ではありえない。また、ある C_i について C_i と A を結ぶ辺と C_i と B を結ぶ辺の両方が G にあれば C_i がAB型であることも分かる。

もし A の点と B の点を結ぶ辺が G に存在すれば、 E_2 の選び方より G_3 にも存在し、 $G_3 - \{v_1, v_2\}$ は連結である。したがって G 上で A の点と B の点を結ぶ辺はないとしてよい。この場合でも G は3-連結であるから、少なくとも1個の C_i は G 上で A の点に隣接する点と B の点に隣接する点の両方 (同じ点かもしれない) をもつ。よってこの C_i はAB型であ

るので、 $G_3 - \{v_1, v_2\}$ は連結であり、仮定に反する。

(証明終)

$|E_1 \cup E_2 \cup E_3| \leq 3n - 5$ であることを示そう。 $G_1 = (V, E_1)$ は G の全域木であるから $|E_1| = n - 1$ である。 $top(v)$ と $e(v)$ は G_1 の根とその子については定義されないので $|E_2|, |E_3| \leq n - 2$ である。したがって、 $|E_1 \cup E_2 \cup E_3| \leq 3n - 5$ である。

E_1 と E_2 が $O(m)$ 時間で求められることは容易にわかる。以下では E_3 を求める方法を示す。

procedure E3;

begin

for 各点 v do $e(v) := undefined$;

for $dfnum$ の降順に各点 u do

for u と子でない子孫 w を結ぶ各辺 (u, w) do

for G_1 において w と u の孫 $x \in ANC(w)$

を結ぶ道の各点 v do

if $e(v) = undefined$ then $e(v) := (u, w)$

end;

手続き E3 で $e(v)$ が正しく求められることは容易に分かるだろう。たどられる道の長さは $O(n)$ であり、高々 m 本の道がたどられるので、計算時間は $O(mn)$ である。

いわゆる path compression¹⁾ の手法を用いて手続き E3 の計算時間を短縮することができる。“ w と u の孫 x を結ぶ道の各点”をたどった後でそれらの全ての点を1点に同一視し、同じ点を数多くたどらないようにすればよい。このような path compression は、disjoint set union algorithm^{7), 10)} を用いることによって高速にインプレメントすることができる。以下ではこれらの手法を用いて手続き E3 の計算時間を $O(m)$ に改善する方法を示す。

disjoint set union algorithm は以下の三つの演算の組合せで互いに素な集合の族を操作するものである。

(1) $makeset(v)$: 一つの要素からなる集合 $\{v\}$ をつくり、その代表を v とする。

(2) $union(v, w)$: 代表が v の集合と代表が w の集合の和集合をつくり、その代表を v とする。もとの二つの集合は除去される。

(3) $find(v)$: v を含む集合の代表を返す。

要素数が n 個の場合に上の三つの演算の任意の r 個の組合せは $O(\alpha(r, n))$ 時間で実行できることが知られている¹⁰⁾。ここで $\alpha(r, n)$ は Ackerman 関数の逆関数である。また、 n 個の要素を点集合とする木 (un-

ion tree) が定義できて, *union* はその二つの集合の間に辺があるときにだけ実行する特別な場合には, r 個の演算が $O(r)$ 時間で実行できることも知られている⁷⁾.

上の(1)~(3)の演算で path compression を実現することができる. 手続き E3 を以下のように書き換える.

procedure E3';

begin

for 各点 v **do** *makeset* (v);

for *dfnum* の降順に各点 u **do**

for u と u の子でない子孫 w を結ぶ各辺 (u, w) **do**

begin

$v := \text{find}(w)$;

{ w からその先祖へ行く道をたどったとき, v は $e(v)$ が未定義の最初の点}

while $p(v)$ は u の子孫 **do**

begin

$e(v) := (u, w)$; { $e(v)$ を定義する}

$y := \text{find}(p(v))$; { $p(v)$ から u へ行く道をたどったとき y は $e(v)$ が未定義の最初の点}

union (y, v); {和集合をとる}

$v := y$

end

end

end;

手続きの実行中いつでも次の(a)~(c)が成立している.

(a) 集合族は V の分割である.

(b) 各集合において代表の *dfnum* はその要素のうちで最小である (代表以外の要素は代表の子孫である).

(c) 各集合について, その代表 v の $e(v)$ だけが未定義である.

このことに注意すれば, 手続き E3' の正当性は容易に証明できる. *union* が高々 $n-1$ 回しか実行されないことに注意すれば disjoint set union の演算は $O(m)$ 回しか実行されないことが分かる. しかも上で述べた特別な場合であるので, 手続き E3' の計算時間は $O(m)$ である.

以上により 3-連結グラフの 3-連結全域部分グラフで辺数が $O(n)$ 本のもが線形時間 $O(m)$ で求められることが分かった.

6. むすび

4. のアルゴリズム PART 3 は $O(mn)$ 時間で 3-連結グラフを分割する. また 5. のアルゴリズム E3' は任意の 3-連結グラフ G の 3-連結全域部分グラフで辺数が $O(n)$ 本のグラフ G_3 を線形時間で見つける. そこで G のかわりに G_3 にアルゴリズム PART 3 を適用するようにすれば, PART 3 は $O(n^2)$ 時間で終了する.

4 分割あるいは一般に k 分割の問題については多項式時間アルゴリズムはまだ知られていない. このようなアルゴリズムを見つけることが今後の課題である.

謝辞 日頃, ご討論いただく東北大学工学部斎藤伸自教授および東京工業大学工学部梶谷洋司教授に感謝する. 本研究は文部省一般研究 (C)62550253 から一部援助を受けた.

参考文献

- 1) Aho, A. V., Hopcroft, J. E. and Ullman, J. D.: The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, MA (1974).
- 2) Bollobás, B.: Extremal Graph Theory, Academic Press, London (1978).
- 3) Dyer, M. E. and Frieze, A. M.: On the Complexity of Partitioning Graphs into Connected Subgraphs, Discrete Appl. Math., Vol. 10, pp. 139-153 (1985).
- 4) 榎本彦衛: グラフ学入門, 日本評論社 (1988).
- 5) Györi, E.: On division of connected subgraphs, Combinatorics (Proc. Fifth Hungarian Combinatorial Coll., 1976, Keszthely), Bolyai-North-Holland, pp. 485-494 (1978).
- 6) 今瀬, 真鍋: ネットワークにおける障害耐力のある固定ルーティング方式について, 電子情報通信学会コンピュータセッション研究会資料, COMP 86-70, pp. 95-105 (1987).
- 7) Gabow, H. N. and Tarjan, R. E.: A Linear-Time Algorithm for a Special Case of Disjoint Set Union, Journal of Computer and System Sciences, Vol. 30, pp. 209-221 (1985).
- 8) Lovász, L.: A Homology Theory for Spanning Trees of a Graph, Acta Math. Acad. Sci. Hungar., Vol. 30, pp. 241-251 (1977).
- 9) 永持, 茂木: k - (辺) 連結全域部分グラフを求めるアルゴリズム, 情報処理学会アルゴリズム研究会資料, 89-AL-10 (1989).
- 10) Tarjan, R. E.: Data Structure and Network Algorithms, pp. 23-31, Soc. Ind. Appl. Math., Philadelphia, PA (1983).

(平成元年 5 月 31 日受付)