

3-連結グラフの3分割アルゴリズム

上野修一* 鈴木均**
高橋奈穂美** 西関隆夫** 宮野浩*

*東京工業大学 **東北大学

本報告では、与えられた3-連結グラフを、指定した点を含みかつ指定した大きさの3つの連結部分グラフに分割する $O(n^2)$ 時間のアルゴリズムを与える。ここで n はグラフの点数である。

An Algorithm for Tripartitioning 3-Connected Graphs

Shuichi Ueno* Hitoshi Suzuki**
Naomi Takahashi** Takao Nishizeki** Hiroshi Miyano*

**Tokyo Institute of Technology*
2-12-1 O-okayama, Meguro-ku, Tokyo 152, Japan

***Tohoku University*
Sendai-shi, Miyagi 980, Japan

In this report, we present an algorithm for partitioning a 3-connected graph G into three connected subgraphs each containing a specified vertex and having a specified size. The algorithm runs in $O(n^2)$ time, where n is the number of vertices in G .

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 $V = V(G)$ と書くことがある。本文ではグラフ G が 3-連結であるときに, グラフ 3-分割問題を解く $O(n^2)$ 時間のアルゴリズムを与える。ここで, n はグラフ G の点数である。またグラフ G の辺数を m と書く。即ち $m = |E|$, $n = |V|$ である。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 V 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) : 以下の (1)~(3) を満足する V の分割 (部分集合の組)

(1) $a_i \in V_i (1 \leq i \leq k)$

(2) $V_i (1 \leq i \leq k)$ によって誘導される G の部分グラフ $G[V_i]$ は連結。

(3) $|V_i| = n_i (1 \leq i \leq k)$

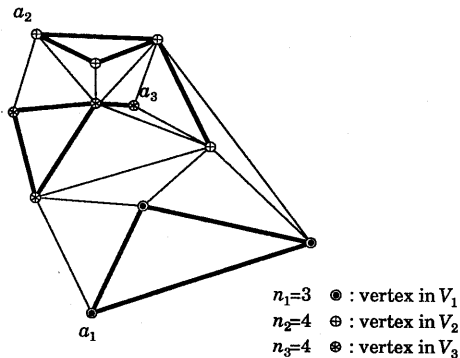


図 1. 3-連結グラフ G とその 3 分割 (太線は $G[V_i]$)

グラフ G が k -連結ならば k -分割問題には必ず解が存在することを Györi[G] と Lovász[L] は独立に証明している。特に $k=2$ の場合は Györi の証明が多項式アルゴリズムを与えている。しかし $k \geq 3$ の場合はいずれの証明も多項式時間アルゴリズムにはなっていない。本文では, $k=3$ の場合の多項式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 [DF]。なおグラフ分割問題は耐障害ルーチングなどに現れる [IM]。

2. 準備

$G = (V, E)$ から $V' (C V)$ の点全て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。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グラフとする。もし $(v, w) \in E$ のときには $G + (v, w) = G$ とする。グラフ G が, $k+1$ 個以上の点を持ちどの $k-1$ 点を取り除いても非連結にならないか, k 点からなる完全グラフであるとき, G は k -連結であるということにする。 G のある辺 (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 -縮約可能であるといい, $G/(v, w)$ 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 [E]。

[定理 1] $n \geq 3$ なる 2-連結グラフ G の任意の点 $v \in V$ には 2-縮約可能な辺が 2 本以上接続している。 ■

3. 2-連結グラフの 2 分割アルゴリズム

本節では 2-連結グラフの 2 分割問題を解くアルゴリズム PART2 と 2 分割問題を少し変形した問題を解くアルゴリズム PART2* を示す。この二つの手続きは次節で示す 3 分割アルゴリズムで用いられる。手続き PART2 は 2-連結グラフ $G = (V, E)$, G の異なる 2 点 a_1, a_2 と $n_1 + n_2 = n = |V|$ なる自然数 n_1, n_2 が与えられたときに前節の (1)~(3) を満足する V の分割 (V_1, V_2) を求める。以下にアルゴリズム PART2 を示す。

function PART2(G, a_1, a_2, n_1, n_2);

begin

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

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

else $\{n_1, n_2 \geq 2, n \geq 4\}$

begin

$a \neq a_2$ なる 2-縮約可能な辺 (a_1, a) 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;

$(V_1, V_2) := \text{PART2}(G, a_1, a_2, n_1 - 1, n_2)$;

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

end

end;

手続き PART2 が正しく V の分割 V_1, V_2 を求めることは定理 1 から容易にわかる。PART2 の計算時間について示そう。2-縮約可能な辺 (a_1, a) を見つけるには、 a_1 に隣接する a_2 以外の点で $G - \{a_1\}$ の切断点ではないものを見つければよい。グラフの全ての切断点は深さ優先探索を 1 回行うことで見つえられる。したがって、2-縮約可能な辺 (a_1, a) は $O(m)$ 時間で見つえられる。また辺 (a_1, a) の縮約の操作は明らかに $O(n)$ 時間で行える。したがって、PART2 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 $O(n)$ 回しか行われないので、PART2 全体の実行時間は $O(mn)$ である。

手続き PART2* は PART2 と同様に 2-連結グラフの 2 分割を求める。ただし、 $n_1 + n_2 = n$ とは限らず、 $n_1 + n_2 \geq n$ であるする。アルゴリズム PART2* の入力と出力は以下の通りである。

(入力)

$G = (V, E)$: 2-連結グラフ

a_1, a_2 : 互いに異なる V の 2 個の点

n_1, n_2 : $n_1 + n_2 \geq n = |V|$ なる自然数

u, w : 互いに異なる V の 2 個の点 ($\{a_1, a_2\} \cap \{u, w\} \neq \emptyset$ 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 $V_1 \cap \{u, w\} \neq \emptyset$

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

以下にアルゴリズム PART2* を示す。

function PART2*($G, a_1, a_2, n_1, n_2, u, w$);

begin

$V_1 := \{a_1\}; V_2 := \{a_2\};$

while $n_1 \geq 2$ **and** $V_1 \cap \{u, w\} = \emptyset$ **do**

begin

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

$G := G / (a_1, a); \{a_1$ と a を同一視して得られる点を新たに a_1 とする }

$V_1 := V_1 \cup \{a\};$

$V := V - \{a\};$

$n_1 := n_1 - 1$

end;

if $n_1 = 1$ **then return** $(V_1, V - V_1);$

while $n_2 \geq 2$ **and** $V_2 \cap \{u, w\} = \emptyset$ **do**

begin

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

$G := G / (a_2, a);$

$V_2 := V_2 \cup \{a\};$

$V := V - \{a\};$

$n_2 := n_2 - 1$

end;

if $n_2 = 1$ **then return** $(V - V_2, V_2);$

$\{V_1 \cap \{u, w\} \neq \emptyset$ かつ $V_2 \cap \{u, w\} \neq \emptyset$

$n'_1 + n'_2 = |V|$ なる自然数 n'_1 と n'_2 を任意に選ぶ;

{ 例えば $n'_1 := \min\{n_1, |V| - 1\}; n'_2 := |V| - n'_1$ とする }

$(V'_1, V'_2) := \text{PART2}(G, a_1, a_2, n'_1, n'_2);$

return $(V_1 \cup V'_1, V_2 \cup V'_2)$

end;

アルゴリズム PART2* の正当性は定理 1 より明らかである。また、計算時間が $O(mn)$ であることも容易にわかるであろう。

4. 3-連結グラフの 3 分割アルゴリズム

本節では 3-連結グラフの 3 分割アルゴリズム PART3 を示す。その入力と出力は次の通りである。

(入力)

$G = (V, E)$: 3-連結グラフ

a_1, a_2, a_3 : 互いに異なる V の 3 個の点

n_1, n_2, n_3 : $n_1 + n_2 + n_3 = n = |V|$ なる自然数

(出力)

(V_1, V_2, V_3) : 以下の (1) ~ (3) を満足する V の分割

(1) $a_1 \in V_1, a_2 \in V_2, a_3 \in V_3$

(2) $G[V_1], G[V_2], G[V_3]$ が連結

(3) $|V_1| = n_1, |V_2| = n_2, |V_3| = n_3$

function PART3($G, a_1, a_2, a_3, n_1, n_2, n_3$);

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 $G[V_i]$ ($i = 1, 2, 3$) にこれらの辺は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、PART2 を適用する。例えば $n_1 = 1$ ならば、

$(V_2, V_3) := \text{PART2}(G - a_1, a_2, a_3, n_2, n_3);$

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

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 に接続し a_2 や a_3 に接続しない辺 (a_1, v_a) を任意に選ぶ。 (a_1, v_a) が 3-縮約可能ならば (3) へ、そうでない時は (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 PART3 を適用する。即ち、

$(V_1, V_2, V_3) := \text{PART3}(G/(a_1, v_a), a_1, a_2, a_3, n_1 - 1, n_2, n_3);$

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

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。 $G' = G - \{a_1, v_a, v_b\}$ が非連結になる点 $v_b \in V(G) - \{a_1, v_a\}$ を見つける。 $G - \{a_1, v_a, a_2\}$ が非連結ならば a_2 を、 $G - \{a_1, v_a, a_3\}$ 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)。以下の (5a) ~ (5c) の 3 通りがある。

(5a) $|V(H) - \{a_1, a_3\}| \leq n_2$ 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) を加えたグラフ $J' = J + (v_a, a_3)$ (図 4 参照) に PART3 を適用する。即ち

$V_2 := V(H) - \{a_1, a_3\};$

$(V_1, V_2', V_3) := \text{PART3}(J', a_1, v_a, a_3, n_1, n_2 - |V_2| + 1, n_3);$

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

を実行して終了する。なお J' は 3-連結であり、また付加した辺 (v_a, a_3) は結果に不都合を生じさせない。

(5b) $|V(H) - \{a_1, a_3\}| > n_2$ 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。即ち

$(V_1, V_2) := \text{PART2}(H - a_3, a_1, a_2, |V(H - a_3)| - n_2, n_2);$

を実行する。もし $v_a \in V_1$ になれば

$(V_1', V_3) := \text{PART2}(J/(a_1, v_a), a_1, a_3,$

$|V(J/(a_1, v_a))| - n_3, n_3);$

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

を実行する。もし $v_a \in V_2$ ならば

$(V_1', V_3) := \text{PART2}(J - v_a, a_1, a_3, |V(J - v_a)| - n_3, n_3);$

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

を実行する。ここで $H - a_3, J/(a_1, v_a), J - v_a$ のいずれも 2-連結であることに注意されたい。

(5c) (5a) でも (5b) でもない時。いま $|V(H - a_3)| > n_1 + n_2$ である。この場合は

$V_3 := V(J) - \{a_1, v_a\};$

$(V_1, V_2, V_3') := \text{PART3}(H + (v_a, a_3), a_1, a_2, a_3,$
 $n_1, n_2, |V(H)| - n_1 - n_2);$

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

を実行する。なお $H + (v_a, a_3)$ は 3-連結であり、辺 (v_a, a_3) は結果に不都合を生じさせない。

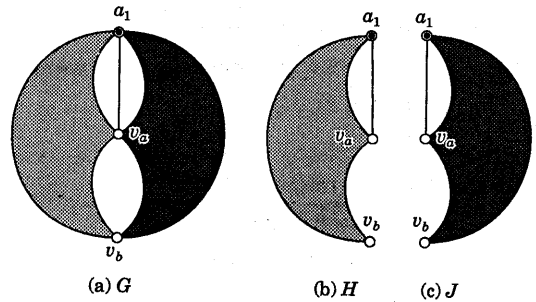


図 2. グラフ (a) G , (b) H , (c) J

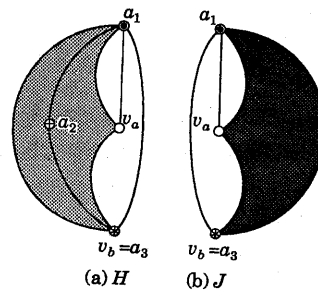


図 3. $v_b = a_3$ の場合のグラフ (a) H と (b) J

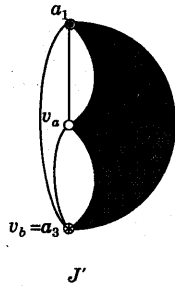


図4. グラフ J

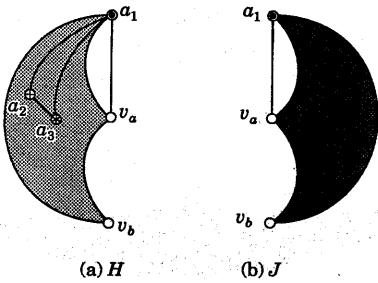


図5. $v_b \neq a_2, a_3$ の場合のグラフ (a) H と (b) J

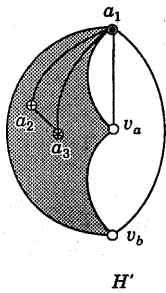


図6. グラフ H'

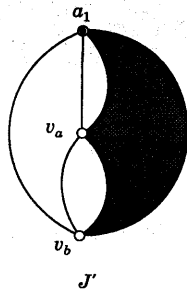


図7. グラフ J'

(6) いま $v_b \neq a_2, a_3$ であり, a_2 と a_3 は隣接しているのととも H に含まれる. (図5参照). $H' = H + (a_1, v_b)$ (図6参照) が3-連結の時は(7)へ, そうでなければ(8)へ行く.

(7) いま H' は3-連結である.

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

$$V_1 := V(J) - \{v_a, v_b\};$$

$$(V'_1, V_2, V_3) := \text{PART3}(H', a_1, a_2, a_3,$$

$$|V(H')| - n_2 - n_3, n_2, n_3);$$

$\text{return}(V_1 \cup V'_1, V_2, V_3)$

を実行し終了する. なお $V(J) - \{v_a, v_b\}$ の点は全て V_1 に含まれるので, 付加した辺 (a_1, v_b) は結果に不都合を生じさせない.

(7b) $|V(H - a_1)| < n_2 + n_3$ ならば

$$(V_2, V_3) := \text{PART2}^*(H - a_1, a_2, a_3, n_2, n_3, v_a, v_b)$$

を実行する. このあと $J' = J + (a_1, v_b) + (v_a, v_b)$ を作る (図7参照). $v_a \in V_2$ かつ $v_b \in V_3$ ならば

$$(V_1, V'_2, V'_3) := \text{PART3}(J', a_1, v_a, v_b,$$

$$n_1, n_2 - |V_2| + 1, n_3 - |V_3| + 1);$$

$\text{return}(V_1, V_2 \cup V'_2, V_3 \cup V'_3)$

を実行し終了する. なお J' は3-連結であり, 辺 (a_1, v_b) , (v_a, v_b) は結果に不都合を生じさせない. $v_a \in V_3$, $v_b \in V_2$ の場合も同様にすればよい. $v_a, v_b \in V_2$ ならば, $V_3 \cap \{v_a, v_b\} = \emptyset$ なので $|V_3| = n_3$ のはずであるから, このとき

$$J'' := J' / (v_a, v_b);$$

$$(V'_1, V'_2) := \text{PART2}(J'', a_1, v_a (= v_b), n_1, |V(J'')| - n_1);$$

$\text{return}(V_1 \cup V'_1, V_2 \cup V'_2, V_3)$

を実行し終了する. むろん J'' は2-連結である. $v_a, v_b \in V_3$ の場合も同様にすればよい.

(8) いま H' は2-連結である. $H'' = H' - a_1$ は図8のように2-連結成分の鎖になる (補題1参照). 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 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.

(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9参照),

$$(V'_1, V_2, V_3) := \text{PART3}(M, a_1, a_2, a_3, n_F - n_2 - n_3 + 1, n_2, n_3);$$

$\text{return}(V_1 \cup V'_1, V_2, V_3)$

を実行し終了する. なお M は3-連結である (補題2参照). また $V - V(F)$ の点は V_1 に含めているので, 辺 (a_1, u) , (a_1, w) は結果に不都合を生じさせない.

(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 $n_F = 2$, 即ち 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。

$(V_2, V_3) := \text{PART2}^*(F, a_2, a_3, n_2, n_3, u, w)$

を実行する。 $u \in V_2$ かつ $w \in V_3$ または $u \in V_3$ かつ $w \in V_2$ 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。 $G - (V(F) - \{u, w\})$ に辺 $(a_1, u), (a_1, w), (u, w)$ を加えたグラフを B とする (図 9 参照)。

$(V_1, V_2', V_3) := \text{PART3}(B, a_1, u, w,$

$n_1, n_2 - |V_2| + 1, n_3 - |V_3| + 1);$

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

を実行し終了する。 B は 3-連結である。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 $|V_3| = n_3$ である。 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 (図 11 参照),

$(V_1, V_2') := \text{PART2}(B', a_1, u (= w), n_1, |V(B')| - n_1);$

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

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 PART3);

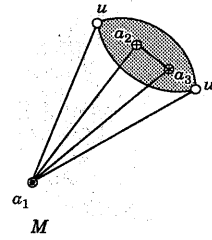


図 9. 3-連結グラフ M

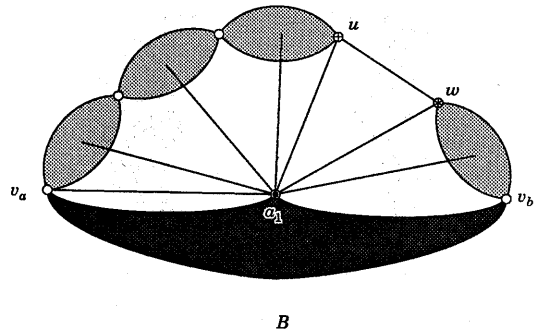
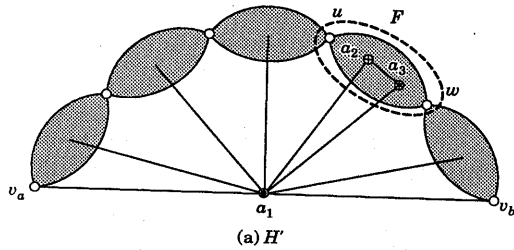
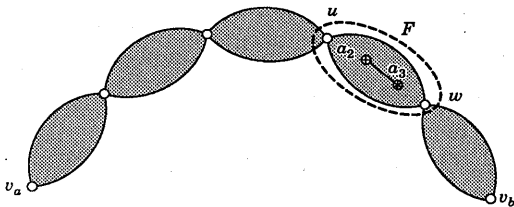


図 10. 3-連結グラフ B



(a) H'



(b) H''

図 8. 2-連結グラフ (a) H' と連結グラフ (b) H''

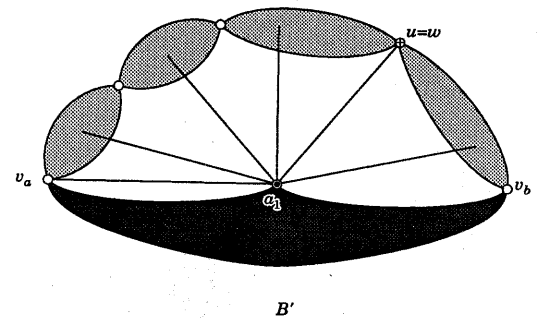


図 11. 2-連結グラフ B'

アルゴリズムの正当性は次の 2 つの補題を用いれば容易に証明できる。

[補題 1] アルゴリズム PART3 の (8) で H'' は連結であり 2-連結ではない。しかも H'' の切断点の 1 つを u_i とすると $H'' - u_i$ は丁

度 2 つの連結成分を持ち、それぞれの成分は v_a または 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 $H' - \{v_1, v_2\}$ の連結成分は v_a か v_b を含む。従って $H' - \{v_1, v_2\}$ は丁度 2 個の連結成分を持っていて、それぞれは v_a か 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。従って、 $H'' = H' - a_1$ 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 1 つを v とすると、 $H'' - v = H' - \{v, a_1\}$ は丁度 2 つの成分をもち、それぞれの成分は v_a または v_b を含む。(証明終)

補題 1 より H'' が 2-連結成分の鎖になっていることがわかる。

[補題 2] アルゴリズム PART3 の (9) で M は 3-連結である。

(証明) M に切断点对 $[v_1, v_2]$ があるすると、 G の 3-連結性から、 $M - \{v_1, v_2\}$ の連結成分は u, w のいずれかを含まなければならない。ところが辺 (a_1, u) , (a_1, w) が存在するので、 a_1 はいずれの成分にも含まれない。よって $a_1 \in \{v_1, v_2\}$ である。このとき v_1 と v_2 のうち a_1 ではない方の点は F の切断点であることになってしまい、 F の 2-連結性に反する。(証明終)

PART3 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 $O(m)$ 時間で終了する。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 $O(m)$ 時間で終了する。(6) で H' が 3-連結かどうかを調べるには $H' - a_1$ が切断点を持つかどうかを調べればよい。したがって (6) は $O(m)$ 時間で終了する。また (6) で、 $H' - a_1$ の全ての切断点を見つか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 $O(m)$ であることがわかる。以上より PART3 を再帰呼び出しする部分と、PART2, PART2* を用いる部分以外に要する時間は $O(m)$ であることがわかる。アルゴリズムは (1) 以外では (3), (5), (7), (9), (11), (12) で終了する。PART2, PART2*, PART3 の適用のしかたには以下の 3 通りがある。

- (A) PART2 と PART2* だけを適用する。((5b),(7b),(12))
- (B) PART3 だけを G より点数が少ないグラフに適用する。((3),(5a),(5c),(7a),(9))
- (C) PART2* 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 PART3 を適用する。((7b),(11))

PART3 は必ず点数が少なくなったグラフに対し再帰呼び出されることに注意されたい。したがって、PART3 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 PART3 が $O(mn)$ 時間で終了することがわかる。

5. むすび

前節のアルゴリズム PART3 は $O(mn)$ 時間で 3-連結グラフを 3 分割する。任意の 3-連結グラフには $O(n)$ 本の辺を持つ 3-連結全域部分グラフ G' が必ず存在する。しかも G' を $O(m)$ 時間で見つけることができる。 G のかわりに G' にアルゴリズム PART3 を適用するにすれば、PART3 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。これらの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、御討論頂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深謝する。本研究は文部省一般研究 (C)62550253 から一部援助を受けた。

文献

- [DF] M. E. Dyer and A. M. Frieze, On the complexity of partitioning graphs into connected subgraphs, *Discrete Appl. Math.*, 10(1985), pp.139-153.
- [E] 榎本彦衛, グラフ学入門, 日本評論社, 1988.
- [G] E. Györi, On division of connected subgraphs, *Combinatorics (Proc. Fifth Hungarian Combinatorial Coll., 1976., Keszthely), Bolyai-North-Holland (1978), pp.485-494.*
- [IM] 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チング方式について, 信学会コンピュータシオン研資, COMP86-70, pp.95-105.
- [L] L. Lovász, A homology theory for spanning trees of a graph, *Acta Math. Acad. Sci. Hungar.* 30(1977), pp.241-251.