3-連結全域部分グラフを求めるアルゴリズム

鈴木均　高橋奈穂美　西関隆夫

東北大学

本報告では3-連結グラフ G の $O(n)$ 本の辺を持つ3-連結全域部分グラフを求める $O(m)$ 時間のアルゴリズムを与える．ここで n，m はそれぞれグラフの点数，辺数である．まず，$G=(V,E)$ の全域木 $G_1=(V,E_1)$ に E に含まれる高々 $n-2$ 本の辺を付け加えて，2-連結グラフ $G_2=(V,E_1\cup E_2)$ を求め，さらに G_2 に高々 $n-2$ 本の辺を付け加えることで3-連結グラフ $G_3=(V,E_1\cup E_2\cup E_3)$ を構成する．また，$G_2'=(V,E_1\cup E_2^{min})$ が2-連結であり，かつ E_2^{min} のどの辺を取り除いてもグラフが2-連結ではなくなってしまうような E_2^{min} を求める $O(m)$ 時間のアルゴリズムも示す．

An Algorithm for Finding a Triconnected Spanning Subgraph

Hitoshi SUZUKI　Naomi TAKAHASHI　Takao NISHIZEKI

Department of Electrical Communications
Faculty of Engineering
Tohoku University
Sendai-shi, Miyagi 980, Japan

Let $G=(V,E)$ be a triconnected graph with vertex set V and edge set E. In this report, we present a linear algorithm for finding a triconnected spanning subgraph G_3 of G with $O(n)$ edges, where n is the number of vertices of G. The algorithm first constructs a biconnected spanning subgraph G_2 of G by adding at most $n-2$ edges of G to a spanning tree $G_1=(V,E_1)$, then G_3 by adding at most $n-2$ edges of G to G_2.

We also present an algorithm for finding E_2^{min} such that the graph $G_2'=(V,E_1\cup E_2^{min})$ is biconnected and every edge in E_2^{min} is essential, that is, for any edge $e\in E_2^{min}$, $G_2'-e$ is not biconnected.

1. まえがき

本文では3-(2-)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-(2-)連結全域部分グラフ $G_3(G_2)$ を求める線形時間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．

グラフの2-連結性の判定，3-連結性の判定は線形時間でできることが知られている[E],[HT]．また一般にグラフの k-連結性の判定が多項式時間でできることも知られている．従って各辺毎にそれを除去して k-連結性の判定を行うことで，与えられた k-連結グラフの極小 k-連結全域部分グラフを多項式時間で求めることができる．また k-連結グラフの最小 k-連結全域部分グラフを求める問題は任意の固定した $k \geq 2$ について NP-完全であることも知られている[GJ]．

本文の3.では G の2-連結全域部分グラフで高々 $2n-3$ 本の辺を持つグラフ G_2を，4.では Eに含まれる高々 $n-2$ 本の辺を G_2 に付加して高々 $3n-5$ 本の辺を持つ G の3-連結全域部分グラフ G_3を求める．5.では，3.で求めた G_2よりさらに辺の少ない2-連結全域部分グラフを求めるアルゴリズムを与える．いずれも線形時間アルゴリズムである．

なお一般に点数 $n \geq 3k-2$ 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[B]．本文で与えるアルゴリズムを用いれば $O(n^2)$ 時間で極小2-, 3-連結全域部分グラフが求まる．

2. 準備

$G=(V,E)$ 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．$G=(V,E)$ 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．辺集合 $E' \subseteq \{(v,w)|v,w \in V\}$ に対し，$G+E'$は G に E'の辺を付加して得られる単純グラフとする．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 $k+1$ 個以上の点を持ちどの $k-1$ 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．連結グラフ G に対し，$G-\{v\}$ が非連結であるとき，点 vは切断点と呼ばれる．G が2-連結であり，$G-\{v_1,v_2\}$ が非連結であるとき，点対 v_1,v_2 は切断点対と呼ばれる．k-連結グラフ $G=(V,E)$ において，辺 $e \in E$を取り除くとグラフが k-連結ではなくなるならば，辺 e はk-エッセンシャルであるという．$G_1=(V,E_1)$ 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自身を含むものとする．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 [AHU] を参照されたい．

3. 2-連結全域部分グラフを見つけるアルゴリズム

$G_1=(V,E_1)$ は G の深さ優先探索木であるから $(v,w) \in E$ならば G_1上で wは vの先祖か子孫である．$G_1=(V,E_1)$ は木であるから(1-)連結である．$dfnum(v)$ 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 ANC$(p(p(v)))$ の点で 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．即ち

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

[補題1] $G=(V,E)$ が2-連結ならば，$G_2=(V,E_1 \cup E_2)$ も2-連結である．

(証明) $G_2-\{v\}$ が非連結になる点 $v \in V$が存在すると仮定する．G は2-連結だから $G-\{v\}$ に DES$(v)-\{v\}$ と ANC$(p(v))$ 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$(v)-\{v\}$ と ANC$(p(v))$ 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．従って G_2は2-連結である． (証明終)

G_1は明らかに $O(m)$ 時間で求められる．また，全ての点 $v \in V$ についての $top(v)$ も $O(m)$ 時間で求められることも容易にわかるだろう．$|E_1|=n-1$ であり，G_1の根 r，rの子 vについての $top(r)$，$top(v)$ は定義されない．よって $|E_1 \cup E_2| \leq 2n-3$ である．なお G が2-連結であれば根 rの子は1個しかないことに注意されたい．

以上により $O(m)$ 時間で高々 $2n-3$ 本の辺を持つ2-連結全域部分グラフを求めることができることがわかる．

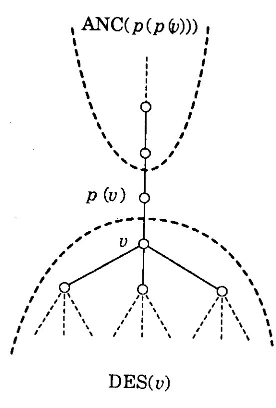

図1. 深さ優先探索木の一部

4. 3-連結全域部分グラフを見つけるアルゴリズム

各点 $v \in V$について ANC($p(p(v))$) と DES(v) (図1参照) を結ぶ辺 (u, w) で $dfnum(u)$ が最大なものの任意の 1 本を $e(v)$ とする (但し $u \in$ANC($p(p(v))$)，$w \in$DES(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．即ち

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

[補題 2] 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_sとする．$V - \{v_1, v_2\}$ を以下のように 3 つの点集合 X_A，X_B，X_Cに分ける (図2参照)．

$X_C =$DES$(v_2) - \{v_2\}$

$X_B =$ DES$(v_s) -$ DES(v_2)

$X_A = V - (X_B \cup X_C \cup \{v_1, v_2\})$

なお，v_sが存在しないとき，即ち $p(v_2) = v_1$のときは $X_B = \phi$であるとする．以下では $X_A, X_B, X_C \neq \phi$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 $C_1, C_2, \ldots, C_l$とする．各 C_iの点集合は v_2 の 1 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 型であることもわかる．

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 (同じ点かも知れない) を持つ．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．前節で示したように $|E_1 \cup E_2| \leq 2n - 3$ であり，また $e(v)$ は G_1の根とその子については定義されないので $|E_3| \leq n - 2$ である．したがって，$|E_1 \cup E_2 \cup E_3| \leq 3n - 5$ である．

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 u の子孫 wを結ぶ各辺 (u, w)(ただし $w \neq u, u$ の子) **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;

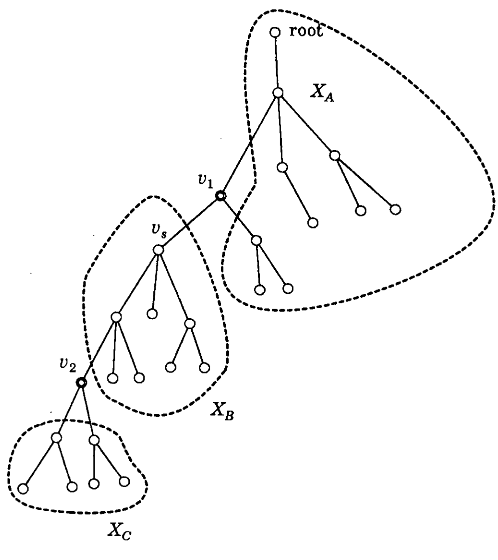

図 2. 補題 2 の証明の説明図

手続き E3 で $e(v)$ が正しく求められることは容易にわかるだろう．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 compression[AHU]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[GT],[T]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 3 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)$makeset(v)$ ：1 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 2 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 3 つの演算の任意の r 個の組合せは $O(r\alpha(r, n))$ 時間で実行できることが知られている[T]．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 (union tree) が定義できて，$union$ はその 2 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[GT]．

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 の孫 w を結ぶ各辺 (u, w)(ただし w ≠ u, u の子)
      do
        begin
          v := 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) ≠ u do
            begin
              e(v) := (u, w); {e(v) を定義する}
              y := 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 $dfnum$ はその要素の内で最小である (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わかる．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 $O(n)$ 本のものが線形時間 $O(m)$ で求められることがわかった．

なお，$G'_2 = G_1 + E_3$ は G の 2-連結全域部分グラフである．

5．2-連結全域部分グラフの辺の極小化

本節では 2-連結グラフ $G = (V, E)$ と G の深さ優先探索木 G_1 が与えられたときに，$G_1 + E_2^{min} (E_2^{min} \subseteq E)$ が 2-連結であり，かつ 2-エッセンシャルな辺だけからなる辺集合 E_2^{min} を求める線形時間アルゴリズム E2minimal を与える．

各点 $v \in V$ について $\mathrm{ANC}(p(p(v)))$ と $\mathrm{DES}(v)$ を結ぶ辺 (u, w) で $dfnum(u)$ が最小なもので $dfnum(w)$ が最大のものを $e'_2(w)$ とする (但し $u \in \mathrm{ANC}(p(p(v)))$，$w \in \mathrm{DES}(v)$ とする)．そのような辺がなければ $e'_2(v)$ を定義しない．$e'_2(v)$ の集合を E'_2 と書く．即ち

$E'_2 = \{e'_2(v) | v \in V$ かつ $e'_2(v)$ が定義される$\}$．

明らかに $E'_2 \subset E_2$ である．E'_2 が $O(m)$ 時間で求められることは容易にわかるだろう．次のアルゴリズム E2minimal では 2-エッセンシャルな辺の集合 $E_2^{min} \subset E'_2$ を求める．

```
procedure E2minimal;
  begin
    E2min := φ;
    for dfnum の降順に各点 v ≠ r, r の子 do
      if DES(v) と ANC(p(p(v))) を結ぶ辺が E2min にない then
        begin
          e2min(v) := e'2(v);
          E2min := E2min ∪ {e2min(v)}
        end
  end;
```

E2minimal の計算時間は明らかに $O(n^2)$ 時間である．

次の補題が成立する．

[補題 3] G が 2-連結ならば手続き E2minimal で求められた E_2^{min} を G_1 に加えたグラフ $G_2^{min} = G_1 + E_2^{min}$ は 2-連結であり，また G_2^{min} 上で E_2^{min} の辺は全て 2-エッセンシャルである．

(証明) G_2^{min}が 2-連結であることは明かである．以下では E_2^{min}の各辺が 2-エッセンシャルであることを示す．E_2^{min}の辺 $e=(v_h,v_l)=e_2^{min}(v)$ に対して $G'=E_1+E_2^{min}-\{e\}$ が 2-連結であると仮定する． ただし $v_l \in$ANC$(p(p(v)))$，$v_h \in$DES(v) とする．

G'は 2-連結であるから，$p(v)$ は切断点ではない．したがって $E_2^{min}-\{e\}$ には ANC$(p(p(v)))$ と DES(v) を結ぶ辺が含まれるはずである．そのような辺の 1 本を $e^*=(x_h,x_l)=e_2^{min}(x)$ とする．手続き E2minimal で e^*が e より先に E_2^{min}に含まれたとすると，e は E_2^{min}の辺として選ばれなかったはずであるから，e^*は e より後で E_2^{min}に含まれたはずである．従って x は vの先祖であり，DES(v) $\subset$DES(x) である．よって e'_2の選び方により $dfnum(x_l) \le dfnum(v_l)$ である．$dfnum(x_l) < dfnum(v_l)$ とすると，e は "ANC$(p(p(v)))$ と DES(v) を結ぶ辺 (u,w) で $dfnum(u)$ が最小なもの" ではなく，したがって $dfnum(x_l)=dfnum(v_l)$ である．同様に e'_2の選び方より $dfnum(x_h) < dfnum(v_h)$ あるいは $dfnum(x_h) > dfnum(v_h)$ のいずれでもありえない．よって $e^*=e$ あり，$E_2^{min}-\{e\}$ が e^*を含むことはなく G'は 2-連結ではない． (証明終)

上のアルゴリズム E2minimal も E3 と同様に disjoint set union algorithm を用いて高速化できる．以下にそのアルゴリズムを示す．

```
procedure E2minimal';
  begin
    E_2^min := φ;
    for 各点 v do makeset(v);
    for dfnum の降順に各点 v ≠ r, rの子 do
      begin
        if v = find(v) then
          begin
            e_2^min(v) := e'_2(v);
            e_2^min(v) の ANC(p(p(v))) にある端点を v_l とする;
            E_2^min := E_2^min ∪ {e_2^min(v)}
            u := v;
            while dfnum(p(u)) > dfnum(v_l) do
              begin
                x := find(p(u));
                union(x, u);
                u := x
              end
          end
      end
  end;
```

手続きの実行中いつでも以下の (a) ～ (c) が成立する．

(a) 集合族は Vの分割である．

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

(c) 各集合について，その代表の点 vについてだけ DES(v) と ANC$(p(p(v)))$ を結ぶ辺がない．

このことに注意すれば E2minimal'の正当性は証明できる．E3' と同様に E2minimal'の計算時間は $O(m)$ 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に深謝する．

文献

[AHU] Aho, A. V., Hopcroft, J. E. and Ullman, J. D.: The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, MA(1974).

[B] Bollobás, B.: Extremal Graph Theory, Academic Press, London(1978).

[E] Even, S.:Graph Algorithms, Computer Science Press, Maryland(1979).

[GT] 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, 30, pp.209-221(1985).

[GJ] Garey, M. R. and Johnson, D. S.:Computers and Intractability, A Guide to the Theory of NP-Completeness. Freeman, W. H. and Company, 1979.

[HT] Hopcroft, J. E. and Tarjan, R. E.: Dividing a graph into triconnected components, SIAM J. on Comput., 2, 3, pp.135-158(1973).

[T] Tarjan, R. E.: Data Structure and Network Algorithms, pp.23-31, Soc. Ind. Appl. Math., Philadelphia, PA(1983).