グラフの連結性判定アルゴリズム
Algorithms for connectivity test of graphs

数学専攻　池知　佐紀
IKECHI Saki

1 始めに

電力を供給するときに都市を点にし、点同士を結ぶケーブルを辺とすると、グラフとして考えることができる. 任意の頂点を除いてもグラフが連結になっていれば電力が安定して供給することができる. つまり k-連結であるようなグラフになっていれば、安定して電力を供給できる. どんなグラフに対しても線形時間で k-連結性を求められるアルゴリズムがあればいいのだが、それはなかなか難しい. 今回は、極大平面的グラフでの 4-連結性判定と平面的グラフの 3-連結性判定のアルゴリズムから k-連結性判定アルゴリズムに拡張する.

2 準備

G をグラフとする時 V(G) と E(G) はそれぞれ頂点集合と辺集合とする. 平面的グラフ G とは、平面に埋め込まれた平面的グラフのことである. ここででてくるグラフは全て平面的グラフ G とする.G の面とは、平面的グラフ G で頂点と辺によってある平面をいくつかの連結領域に分割されたものをいう. ただひとつ存在する非有界な面を外面といい、残りの面を内面という. グラフ G での最小の面のことを triangle とし、長さ 3 のサイクルのことである.indegree とは、頂点 v に入る辺の本数で outdegree は頂点 v から出る辺の本数のことである.arc とは、頂点 u,v 対して向きがつけられている辺の事である. X と $\bar{X}$ は、向き付けされたグラフの頂点の部分集合とする.$\omega^+(X)$ とは、X から $\bar{X}$ への向き付けされた辺の集合であり、$\omega^-(X)$ とは、$\bar{X}$ から X への向き付けされた辺の集合とする.cocycle $\omega(X)$ とは、$\omega^+(X), \omega^-(X)$ の和集合のことである.cocycle $\omega(X)$ が elementary であるとは、G_X と $G_{\bar{X}}$ が連結している時のことである.$\omega(X)$ =$\omega^+(X)$ ならば、cocycle $\omega(X)$ が positive cocircuit である.angle とは、ある頂点で右回りの順に連続した辺のこととする. 2-連結平面的グラフでの angle graph A(G) とは、G の頂点と面の集合の接続グラフです.(それぞれを $A(G)$ の $V-vertex$ と $F-vertex$ とする.)$A(G)$ の辺と G の angle の辺は一致し、その数は、G の辺の 2 倍です. また、separating triangle とは、3 頂点と接続している辺を全て除くと元のグラフを非連結にしてしまうような triangle のことである.

3 極大平面的グラフでの 4 − 連続性のテスト

アルゴリズムは次の特性に基づいている

- 極大平面的グラフは 4-連結である $\Longleftrightarrow$separating triangle を持たない. つまり、それぞれの triangle が面となる
- どんな極大平面的グラフも全ての頂点が indegree 3 (3 つの外側の頂点は除く) をもつように向きづけされている.
- そのような向き付けの中で、separating triangle は G の positive cocircuit に一致する.

Lemma3.1
G を $3-$ 連結平面的グラフとし、{x,y,z} は G の $cutset$ とする.
その時, $G-$ {x,y,z} は 2 つの連結成分を持つ.
Lemma3.2
極大平面的グラフの triangle は separating triangle である $\Longleftrightarrow$ triangle が面ではない

Lemma3.3

極大平面的グラフ G が 4 －連結 $\Longleftrightarrow$ separating triangle をもたない. つまり、cutset が triangle の辺集合である.

Lemma3.4

G を全ての頂点が *indegree*3 をもつように向き付けされた極大平面的グラフとする.(少なくとも 5 個の頂点から成る) ただし、*indegree* 1 の外面の頂点 3 個は除く.

そのとき、G が 4 －連結である $\Longleftrightarrow$ G が唯一つの positive cocircuit をもつ. つまり、それは、外面の頂点集合によって定義される.

Algoritm 1 A 4-connexity test for a maximal planar graph G

Require: G は極大平面的グラフとする

Ensure: *IsFourConnected*=true $\Longleftrightarrow$G は 4-連結

```
if G は 6 個以下の頂点をもつ then
    IsFourConnected ← false
else
    G'←G
    r1, r2, r3 ←G' のある面の頂点
    全ての頂点が indegree3 をもつように G' を向き付けする (indegree1 をもつ r1, r2, r3は除く)
    頂点 r1, r2, r3 を消す
    G' の向き付けされた双対グラフ H を計算する
    if H の向きが acyclic である then
        IsFourConnected ← true
    else
        IsFourConnected ←false
    end if
end if
```

定理 3.5

極大平面的グラフが 4 – 連結であるかどうかをアルゴリズム 1 が線形時間でテストする

4　平面的グラフの triangle の数え上げ

定義 4.1　Schnyder 分解

G を極大平面的グラフとし、$\{r_1, r_2, r_3\}$ をグラフ G の面の中の一つとする. $\{r_1, r_2, r_3\}$ に関する Schnyder 分解は、G の辺の三色塗りわけです.その塗り分け方は、G の頂点集合で 3 つの全順序 $<_1, <_2, <_3$ を r_i を根とした向き付けされた木 Y_i で満たすようにそれぞれの色を 1≦i≦3 とし、次を満たすように定義する.

- arc(u, v) が Y_i に属し、$(u <_j v)$ になる.$\Longleftrightarrow$　$(j \neq i)$
- $\{x$,y$\}$ が G の辺ならば、その時 ∀u $\notin \{x, y\}$, ∃1≤ i ≤ 3,　$u >_i x$ と $u >_i y$

定義 4.2

G をn≥3 の頂点をもつ平面的グラフとし、r_1, r_2, r_3 を G の 3 つの頂点とする.$\{r_1, r_2, r_3\}$ に関する parent triplet(π_1, π_2, π_3) は G に辺を加え、全ての面が triangle になるように H を作り、$\{r_1, r_2, r_3\}$ に関する H の Schnyder 分解を満たすような V(G) から V(G)∪{0} の triplet 関数とする. その関数 $\pi_i(v)$ は、G でこれらの頂点が隣接しているならば頂点 v の親になり、そうでない場合は 0 になるように定義する.

定理 4.1

平面的グラフ G の *parent triplet* は アルゴリズム 2 を使って線形時間で計算されるかもしれない.

Lemma4.2

色 i の辺の向きを逆にする時、そのグラフは、*acyclic* になる.

Lemma4.3

G の *triangle* が *circuit* である $\Leftrightarrow$ *triangle* が 3 色に塗り分けられる, そうでなければ *triangle* は 2 色に塗り分けられる

Algoritm 2 Computation of a parent triplet

Require: G は少なくとも 4 個の頂点を持つ *planar* グラフとする

Ensure: π_1, π_2, π_3 は r_1, r_2, r_3 に関する *Schnyder parent* 関数とする.

$H \leftarrow G$
G で *triangle* になるように辺を加え、加えた辺にしるしをつける
$r_1, r_2, r_3 \quad \leftarrow \quad H$ のある面の辺
parent function π_1, π_2, π_3 として H の *Schnyder orientation* を計算する ($\pi_i(0) = 0$ と拡張して)
$\pi_1(r_2) \leftarrow r_1$
$\pi_2(r_3) \leftarrow r_2$
$\pi_3(r_1) \leftarrow r_3$
for all しるしをつけた辺 $e = \{u,v\}$ **do**
    for all $i \in \{1,2,3\}$ **do**
        if $u = \pi_i(v)$ **then**
            $\pi_i(v) \leftarrow 0$
        else if $v = \pi_i(u)$ **then**
            $\pi_i(v) \leftarrow 0$
        end if
    end for
end for

定理 4.4

アルゴリズム は平面的グラフの *triangle* を線形時間で数え上げる.

Algoritm 3 Enumeration of the triangles of a planar graph

Require: G は少なくとも 4 個の頂点を持つ平面的グラフとする.

Ensure: *NumberOfTriangles* は G の *triangle* の個数とする

G の *Schnyder parent function* π_1, π_2, π_3 を計算する
NumberOfTriangles $\leftarrow 0$
for all 頂点 v **do**
    for all $(i,j) \in \{1,2,3\}^2, i \neq j$ **do**
        if $(\pi_i(v) \neq 0)$ かつ $(\pi_j(v) \neq 0)$ かつ $(\pi_i(\pi_j(v)) = \pi_i(v))$ **then**
            $NumberOfTriangles \leftarrow NumberOfTriangles + 1$
        end if
    end for
    if $(\pi_1(v) \neq 0)$ かつ $(\pi_2(\pi_1(v)) \neq 0)$ かつ $(\pi_3(\pi_2(\pi_1(v))) = v)$ **then**
        $NumberOfTriangles \leftarrow NumberOfTriangles + 1$
    end if
    if $(\pi_1(v) \neq 0)$ かつ $(\pi_3(\pi_1(v)) \neq 0)$ かつ $(\pi_2(\pi_3(\pi_1(v))) = v)$ **then**
        $NumberOfTriangles \leftarrow NumberOfTriangles + 1$
    end if
end for

Lemma4.5

(a,b,c,d) を C_4 とする. そのとき、(a,b) と (c,d) が同じ木 Y_i の *arc* になることはできない.

5 平面的グラフでの 3 − 連結テスト

このアルゴリズムは、下に続く特性に基づいている.

- 2 − 連結平面的グラフが 3 − 連結 $\Longleftrightarrow$ そのグラフの *angle* グラフの C_4 がそれぞれ面である.
- *planar quadragulation* はすべての頂点が *indegree*2 をもつように向きをつける. *indegree*1 をもつ 4 個の外部の頂点は除く.
- そのような向きの中で面ではない C_4 は *positive cocircuit* に一致する.

Algoritm 5 3-connexity test for a 2-connected planar graph G

Require: G は 2− 連結平面的グラフである

Ensure: $x = true \iff G$ が 3− 連結

```
if G は 4 個の頂点より少ない then
    x ← false
else
        H ← A(G)
        b_1, w_1, b_2, w_2 ←H の面の頂点
    全ての頂点 (b_1, b_2 を除く) が 2 個の incoming edge をもつように H を向き付けする
    頂点 b_1, w_1, b_2, w_2 を除く
    D ← 向き付けされた H の双対グラフ
    if D が連結していてその向きが acyclic である then
        x ← true
    else {D は向きのついた circuit をもつ }
        x ← false
    else if
else if
```

Lemma5.1

G を $2-$ 連結平面的グラフとする.その時、G が 3− 連結 $\iff$ $A(G)$ の C_4 が面である

Remark 5.2

$3-$ 連結平 面的グラフの C_4 を数え上げるアルゴリズムは線形時間ではない. しかしアルゴリズム *6* を利用して C_4 をつくる頂点のリストをあげることはできる.

Lemma 5.3

G は少なくとも 4 個の頂点をもち、2− 連結平面的グラフとする.$A(G)$ はその *angle graph* とする.*indegree*2 をもつように向き付けし、*indegree*1 をもつ外面の頂点は除く.

その時、グラフ G が $3-$ 連結 $\iff$ $A(G)$ が唯一つの *positive cocircuit* をもつ. 言い換えると、その *positive cocircuit* は外面の頂点集合によって定義される.

Lemma5.4

G を 2− 連結平面的グラフとし、e_0 を G の辺とする.r_1 と r_3 が $V-vertex$ である場合、$\{r_1, r_2, r_3, r_4\}$ を e_0 に対応する $A(G)$ の面とする.

G のどんな向きも、$A(G)$ の向きで定義される.G の一致した *angle* が *extermal* ならば $A(G)$ の辺は *incident* された *V-vertex* から *F-vertex* へ向き付けされる. G が e_0*-bipolarly oriented* ならば、$A(G)$ の誘導された向きは、*source* である r_1 と r_3 は除いて全ての頂点が *indegree* 2 をもつように向き付けされる.

定理 5.5

アルゴリズム 5 は $2-$ 連結平面的グラフが $3-$ 連結グラフ かどうかを線形時間でテストする.

以上を本論では詳しく述べている.

6 終わりに

連結性について、紹介されているアルゴリズムを利用して k -連結性を判定できるアルゴリズムに拡張したかったが、条件が限定されていたため拡張できなかった. どんなグラフに対しても k -連結性が判定できるアルゴリズムが検証されることを期待する.

参考文献

[1] N.Chiba and T.Nishizeki, *Arboricity and subgraph listing algorithms SIAM J.Computing vol.14(1985),210-223*

[2] H.de Fraysseix and P.Ossona de Mendez, *Connectivity of Planar Graphs vol.5,no5,pp.93-105(2001)*