(n-p)-点連結性の$O(m+np^{3.5})$時間判定法

伊藤大雄　　横山光雄

豊橋技術科学大学情報工学系
〒441 豊橋市天伯町字雲雀ケ丘1-1
電話: 0532-44-6743　電子メール: ito@tutics.tut.ac.jp（伊藤）

概要　与えられたグラフ$G=(V,E)$、整数$1 \le p \le n$に対し、Gが(n-p)-点連結であるか否かを判定し、(n-p)-点連結である場合にはその点連結度も計算する$O(m+np^{3.5})$時間のアルゴリズムを与える（但し$n=|V|$, $m=|E|$）。また、与えられたグラフの点連結度を計算する$O(m+n(n-\kappa(G))^{3.5})$時間のアルゴリズムも与える。これら2つのアルゴリズムはそれぞれ同じ計算時間のオーダーで、与えられたグラフの補グラフG^cに関して(n-p)-点連結性の判定や点連結度の計算問題を解くアルゴリズムにも拡張できる。

An $O(m+np^{3.5})$ time algorithm for determining a given graph is (n-p)-node-connected

ITO Hiro and YOKOYAMA Mitsuo

Department of Information and Computer Sciences
Toyohasi University of Technology
Toyohasi-si Tenpaku-tyo Hibarigaoka 1-1, 441, Japan
Tel.: +81 532 44 6743, E-mail: ito@tutics.tut.ac.jp (ITO Hiro)

Abstract: For a given graph $G=(V,E)$ and a given integer $1 \le p \le n$, an $O(m+np^{3.5})$ time algorithm for determining whether or not G is (n-p)-node-connected and also calculating the node-connectivity when G is (n-p)-node-connected is presented, where $n=|V|$ and $m=|E|$. For calculation of the node-connectivity of a given graph G, an $O(m+n(n-\kappa(G))^{3.5})$ time algorithm is also presented. These two algorithms can be extended to the same problems on the complement graph of a given graph in the same time-complexity, respectively.

1．はじめに

グラフの連結度の判定や計算は重要な問題であり、多くの研究成果がある[1]。連結度には点連結度と枝連結度があるが、本稿では点連結度に着目し、点連結度が高い場合に高速に動くアルゴリズムを与える。すなわち、グラフGの(n-p)-点連結性を判定する問題を考え、$O(m+np^{3.5})$時間で動作するアルゴリズムを与える。但しnは節点数、mは枝数である。本アルゴリズムはグラフGの点連結度$\kappa(G)$がn-p以上である場合には、$\kappa(G)$も同時に求めることが出来る。

グラフの連結度の重要な応用として通信網の耐故障性の評価がある。通信網は交換局を節点に通信リンクを枝に対応させることによってグラフにモデル化できる。通信網をモデル化したグラフがk-点（枝）連結であるとは、その通信網が、k個未満の交換局（通信リンク）の故障に対しては、故障していない点間に関する通信を保証するということを意味する。公共性の高い通信網ほど耐故障性が厳しく要求され、網設計や運用の際には連結度は重要な指針となっている。

近年通信網は従来の電話通信中心のものからマルチメディア通信網、すなわち音声のみならず画像、それも動画、データなどの通信を一般家庭で行い、娯楽、販売、医療など様々な分野で通信網を使用する方向になりつつあり、より生活に密着したものになってきている。そうなると通信網の断絶の与える影響はより深刻となり、どんな場合にもけっして断絶しない通信網が要求されることとなるであろう。網設備の故障はいくら設備の信頼度を上げても避けられないことであり、まして大地震の様な大災害においては広範囲の設備が一度に故障していまう可能性も十分ある。そういった場合には無線通信が有力な救済手段であるが、帯域確保の問題もあり、無線のみに頼るのでは十分ではない。従来は通信網の連結度は２～３といった低い連結度で設計されることが多かったが、以上のことを考慮すると、はるかに高い連結度が要求される様になると思われる。

これまで連結度の判定、計算の問題に関して様々な研究がされ、多くの成果が出されている。まず与えられた無向単純グラフ$G=(V,E)$、２点$x,y\in V$、整数kに対し、x,y間の点連結度$\kappa(x,y)$がk以上であることの判定が$O(\min\{k, \kappa(x,y), n^{1/2}\}m)$時間でできることがEven and Tarjan[2]によって示された。これをそのまま用いれば$\kappa(G)\geq k$の判定は$O(\min\{k, n^{1/2}\}n^2m)$時間でできるが、Even[3]は、高々$k^2+n$個の異なる$x,y\in V$組に対して$\kappa(x,y)\geq k$の判定を行うことによって、$\kappa(G)\geq k$の判定を行うアルゴリズムを提案した。これによって$\kappa(G)\geq k$の判定の計算時間は$O((k+n^{1/2})kn^{1/2}m)$に改善された。その後Galil[4]によってこのアルゴリズムを改良して、$\kappa(G)$の計算を$O((\kappa(G)+n^{1/2})\kappa(G)n^{1/2}m)$時間で行う方法が示された。また、以上のアルゴリズムには全て永持と茨木[5]の工夫が適用でき、グラフが密な場合の計算時間が改善される。すなわち$\kappa(G)\geq k$の判定は$O(m+k^3n^{1.5}+k^2n^2)$時間で、$\kappa(G)$の計算は$O(m+\kappa(G)^3n^{1.5}+\kappa(G)^2n^2)$時間でできる。

しかしこれらのアルゴリズムは基本的に低い連結度を想定して設計されているため、高い連結度の判定、計算には適していない。則ち、$k=\theta(n)$の場合にはk-点連結性の判定には$O(n^{4.5})$時間かかってしまう。この理由は、これまでは低い連結度しか実用上必要なかった、という背景以外に、大部分のアルゴリズムはネットワークフローにおけるフロー

増大法準拠のアルゴリズム、すなわち２点間の互いに素な路を一つずつ構築していくことにより、まず1-点連結であることが判定され、次に2-点連結性が判定され、ということを続け、k-点連結性の判定が行なわれるアルゴリズムであることに原因がある。この方法に従えばグラフGの点連結度を$\kappa(G)$とすると、$\kappa(G)<k$の場合にはk-点連結性の判定を行えば同時に$\kappa(G)$の値も求められるという利点もある。（但し、Even[3]のアルゴリズムは高速化のためにグラフを変形して計算するため、そのままでは$\kappa(G)<k$の場合でも$\kappa(G)$の値は求まらない。しかし$k=1,k=2,k=2^2,\ldots$と倍々にしていく手法を用いることにより計算量を増やすこと無く$\kappa(G)$の値を求めることが可能である[4][6]。）この手法はkが小さい場合に効果的な手法であるが、kが大きい場合にはあまり適さない。従って$0\le\kappa(G)\le n-1$であるから、kが0よりもむしろn-1に近い場合には、逆向きのアプローチが適している様に思われる。

本稿では与えられたグラフGの補グラフG^cを考え、Gがk-点連結であることと同値なG^c上の性質を考えることによって、$O(m+np^{3.5})$時間でGの(n-p)-点連結性を判定するアルゴリズムを得た。本アルゴリズムはグラフGの点連結度$\kappa(G)$がn-p以上である場合には、$\kappa(G)$も同時に求めることが出来る。このアルゴリズムはpが定数の場合には線形時間で動作する。既存のk-点連結性判定アルゴリズムを用いた場合、定数pに対する(n-p)-点連結性の判定には$O(n^{4.5})$時間かかったことを考えるとこれは大幅な時間量削減である。また、$0<c<1$である任意の定数cに対する$\lfloor cn \rfloor$-点連結性の判定は、本アルゴリズムはこれまでの最高速のアルゴリズムと同じ$O(n^{4.5})$時間を達成している。

本アルゴリズムを応用することによって、$\kappa(G)$を計算する$O(m+n(n-\kappa(G))^{3.5})$時間のアルゴリズムも与える。また、グラフGの補グラフG^cの(n-p)-点連結性の判定を行う$O(m+np^{3.5})$時間のアルゴリズムと、$\kappa(G^c)$を計算する$O(m+n(n-\kappa(G^c))^{3.5})$時間のアルゴリズムについても説明する。補グラフ上の高速アルゴリズムについてはあまり研究されておらず、一部有効なアルゴリズムが報告されているのみである[7][8]。

2　諸定義

$G=(V,E)$を無向単純グラフ、節点数をn、枝数をmとする。グラフGの節点集合を$V(G)$、枝集合を$E(G)$と表すこともある。

$W\subseteq V$に対し$E(W;G)=\{(x,y)\in E \mid x,y\in W\}$とし、Wで誘導される部分グラフ$G(W)$を$(W, E(W;G))$で定義する。また、$G-W$を$(V-W, E(V-W;G))$とする。$G-W$が非連結であるとき、$W\subseteq V$は点カットと呼ばれる。グラフ$G=(V,E)$と点カットW、$G-W$の異なる連結成分に属する２節点$x,y\in V$に対し、Wはxとyを分離すると言う。枝を共有しない（すなわち$(x,y)\notin E$である）２節点$x,y\in V$に対し、xとyを分離する点カットの大きさ（節点数）の最小値をx,yの点連結度とし、$\kappa(x,y;G)$で表す。$0\le k\le\kappa(x,y;G)$である整数kに対しx,yはk-点連結であると言う。グラフGの点連結度を$\kappa(G)=\min_{x,y\in V}\kappa(x,y;G)$で定義する。但しGが完全グラフの場合には特に$\kappa(G)=n-1$とする。$0\le k\le\kappa(G)$である整数kに対しGはk-点連結であると言う。

$G^c=(V,E^c)$, $E^c=\{(x,y)\mid(x,y)\notin E\}$をGの補グラフと呼ぶ。

$G=(V,E)$、$S,T \subseteq V$に対し$E(S,T;G)=\{(x,y)\in E \mid x\in S,\ y\in T\}$とする。$G=(V,E)$に対しVが二つの部分集合$S,T\subseteq V$（$S\cup T=V$, $S\cap T=\phi$）に分割され、$E(S,S;G)=E(T,T;G)=\phi$となるとき、Gは二部グラフと呼ばれ、$G=((S,T),E)$とも表現される。本稿では$S,T\neq\phi$のものを特に狭義二部グラフと呼ぶ。

$x\in V$の次数を$d(x;G)$、隣接節点集合を$A(x;G)$で表す。なお、各記号"*(*;G)"はGが明らかな場合には";G"を省略する場合もある。

3 問題の性質

以下の問題を考える

【(n-p)-点連結判定問題】

入力：グラフ$G=(V,E)$、整数$1\le p\le n$（但し$n=|V|$）

要請：Gが(n-p)-点連結であるか否かを判定せよ。

まず(n-p)-点連結性についてよく知られている簡単な性質を次に補題として掲げておく。

補題1　グラフ$G=(V,E)$が(n-p)-点連結であるならば$|E|\ge(n-p)n/2$を満たす。□

証明）$\forall x\in V$, $d(x;G)\ge n-p$より明らか。Q.E.D.

次に(n-p)-点連結性と等価な補グラフ上の性質を与える。

補題2　グラフGが(n-p)-点連結である必要十分条件は、補グラフG^cが、$|V(H)|\ge p+1$である狭義完全二部グラフHを部分グラフとして含まないことである。□

証明）定義よりGが(n-p)-点連結でない必要十分条件は、Vが３つの部分集合$S,T,W\subseteq V$（$S\cup T\cup W=V$, $S\cap T=S\cap W=T\cap W=\phi$）に分割され、$|S|,|T|\ge 1$, $|W|<n-p$, $E(S,T;G)=\phi$を満たすことである。これは$H=((S,T),E(S,T;G^c))$が$|V(H)|\ge p+1$を満たす狭義完全二部グラフであることと同値である（図１）。Q.E.D.

補題3　グラフ$G=(V,E)$において、$d(x;G^c)\ge p$である節点$x\in V$が存在するならばGは(n-p)-点連結ではない。□

証明）$S=\{x\}$,$T=A(x;G^c)$とおけば$|A(x;G^c)|\ge p$より、$H=((S,T),E(S,T;G^c))$は$|V(H)|\ge p+1$を満たす狭義完全二部グラフとなり、補題２より証明できる。Q.E.D.

補題２より、グラフの(n-p)-点連結性の判定は補グラフが節点数p+1以上の狭義完全二部グラフを含むかどうかの判定問題に帰着されることが分かる。ここで、任意の$(x,y)\in E^c$を特定し、$(x,y)\in E(H)$である様な狭義完全二部グラフを探すことを考える。ここで

$V(x,y)=A(x;G^c)\cup A(y;G^c)$

$G(x,y)=(V(x,y),\ E(A(x;G^c),A(y;G^c);\ G))$

と定義する。

補題４　$G^c=(V,E^c)$、$(x,y)\in E^c$に対し、G^cの部分グラフHが$(x,y)\in E(H)$を満たす狭義完全二部グラフである必要十分条件は、$G(x,y)$において$V(x,y)-V(H)$がx,yを分離する点カットとなることである。□

証明）$H=((S,T),\ E(H))$をG^cの部分グラフで、$(x,y)\in E(H)$を満たす狭義完全二部グラフであるとする。$(x,y)\in E(H)$より、一般性を失うこと無く$x\in S$かつ$y\in T$とできる。$S\subseteq A(y;G^c)$, $T\subseteq A(x;G^c)$より、任意の$u\in S, v\in T$に対して$(u,v)\in E^c$、すなわち$(u,v)\notin E$である。よって$V(x,y)-V(H)$は、$G(x,y)$においてx,yを分離する点カットとなる。

次に$W\subset V(x,y)$を、$G(x,y)$においてx,yを分離する点カットとする。則ち、$S,T\subset V(x,y)-W$, $S\cup T=V(x,y)-W$, $S\cap T=\phi$, $E(S,T;G(x,y))=\phi$, $x\in S$, $y\in T$を満たすS,Tが存在する。よって$H=((S,T),E(S,T;G^c))$は狭義完全二部グラフでG^cの部分グラフであり、しかも$(x,y)\in E(H)$である。Q.E.D.

以上より、次の定理を得る。

定理１　Gが(n-p)-点連結であることの必要十分条件は、全ての$(x,y)\notin E$に対し、$\kappa(x,y;G(x,y))\geq|V(x,y)|-p$となることである。□

証明）補題２、４より導かれる。Q.E.D.

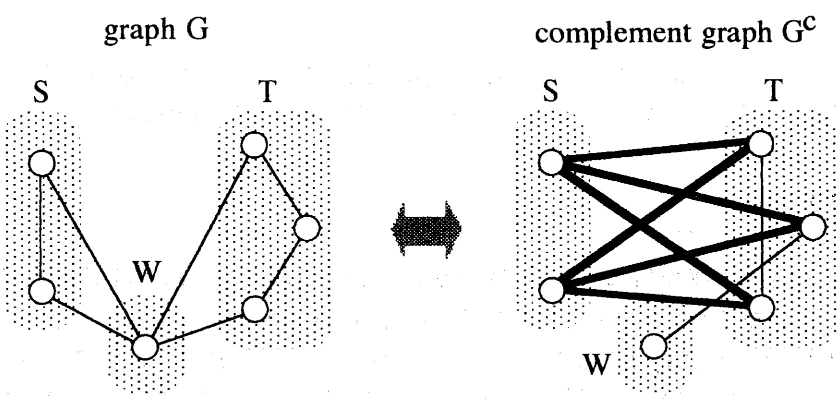

図1. 補題２の説明

If W is a cut separating S and T in G, ((S,T), E(S,T; G^c)) is a strictly bipartite subgraph in G^c.

4　解法

定理１はGの(n-p)-点連結性を判定するアルゴリズムを与える。すなわちGの補グラフ$G^c=(V,E^c)$を作り、各$(x,y)\in E^c$に対して$G(x,y)$を作成し、$\kappa(x,y;G(x,y))\geq|V(x,y)|-p$の判定を行えば良い。ここで、$pmax=\max_{(x,y)\in Ec}\{|V(x,y)|-\kappa(x,y;G(x,y))\}$とすると、Gの連結度はn-pmaxとなることも定理１から容易に導かれる。$G^c=(V,E^c)$を作る前に各節点の次数をチェックし、$d(x)\leq n-p-1$である節点が存在する場合には補題３より(n-p)-点連結でないことが判明し、直ちにアルゴリズムを終了することができる。以上をまとめ、与えられたグラフG=(V,E)、整数$1\leq p\leq n$に対し、Gが(n-p)-点連結であるか否かを判定し、もし(n-p)-点連結であるならばその点連結度κ(G)も同時に与えるアルゴリズムHI-NODE-CONを以下に掲げる。

```
procedure HI-NODE-CON(G=(V,E),p)
begin
1   if there is a node x such that d(x)≤n-p-1
    then output "no"; stop;
2   Construct G^c=(V,E^c); pmax:=1
3   for (x,y)∈E^c do
4      Construct G(x,y);
5      calculate κ(x,y; G(x,y));
6      if κ(x,y; G(x,y))≤|V(x,y)|-p-1
       then output "no"; stop;
       else pmax:=max{pmax, |V(x,y)|-κ(x,y; G(x,y))}
7   enddo
8   output n-pmax;
end.
```

本アルゴリズムの正当性は前述の様に補題３と定理１により保証される。よって次に計算時間を見積もる。第１行目はO(n+m)時間でできる。この操作により、２行目以降では

$$m\geq(n-p)n/2 \tag{1}$$

$$d(x;G^c)\leq p,\ \forall x\in V \tag{2}$$

となる。第２行の操作は$O(n^2)$時間かかるが、$n-p=\Omega(n)$の場合には(1)より$m=\Omega(n^2)$となるので第２行の操作はO(m)時間でできる。(2)より$V(x,y)\leq 2p$となるので、G(x,y)の節点数と枝数は各々O(p)と$O(p^2)$になる。よって４行目の操作は各$(x,y)\in E^c$に対して$O(p^2)$時間で実行可能。節点数n枝数mのグラフの特定節点組間の点連結度は$O(n^{1/2}m)$時間で計算できる[2]ので、第５行の計算は各$(x,y)\in E^c$に対して$O(p^{2.5})$でできる。(2)よりG^cの枝数はO(np)となるので、第３行の繰り返しはO(np)回行えば良い。よって３行から７行の操作は$O(np^{3.5})$時間でできる。第２行の操作は$O(n^2)$時間必要だが、前述の様に$n-p=\Omega(n)$の場合に

はO(m)時間となり、$p=\Omega(n^{2/7})$の場合にも$n^2=O(np^{3.5})$であることを考慮すると、アルゴリズム全体の計算時間は任意のpに対し$O(m+np^{3.5})$となる。以上より次の定理を得る。

定理2　アルゴリズムHI-NODE-CONは、与えられた無向単純グラフG=(V,E)及び整数$1\leq p\leq n$に対して、$\kappa(G)<n-p$であるならば"no"を、$\kappa(G)\geq n-p$であるならば$\kappa(G)$の値を出力する計算時間$O(m+np^{3.5})$のアルゴリズムである。但し$n=|V|$, $m=|E|$とする。□

5．提案算法の応用

5.1. κ(G)の計算アルゴリズム

前節で示したアルゴリズムを応用して、$O(m+n(n-\kappa(G))^{3.5})$時間でκ(G)を計算するアルゴリズムを作ることができる。それはアルゴリズムHI-NODE-CONから１行目の

1　**if** there is a node x such that $d(x)\leq n-p-1$
　then output "no"; **stop**;

の部分を除いただけで実現できる。これをHI-NODE-CON-CALと呼ぶことにする。HI-NODE-CON-CALが正しくκ(G)を計算することは、第４節の始めに書いた様に定理１から保証される。次にこの計算時間が$O(m+n(n-\kappa(G))^{3.5})$で収まることを示す。$D(G^c)=\max_{x\in V} d(x,G^c)$とすると、HI-NODE-CON-CALの計算時間はHI-NODE-CONと同様の考察で$O(m+nD(G^c)^{3.5})$時間となる。ここで補題３より$\kappa(G)<n-D(G^c)$となるのでHI-NODE-CON-CALの計算時間は$O(m+n(n-\kappa(G))^{3.5})$となる。よって以下の定理を得る。

定理3　アルゴリズムHI-NODE-CON-CALは、与えられた無向単純グラフG=(V,E)に対して、κ(G)を計算する、$O(m+n(n-\kappa(G))^{3.5})$時間のアルゴリズムである。□

5.2. 補グラフ上の問題

これまでに示した２つのアルゴリズムHI-NODE-CONおよびHI-NODE-CON-CALでは「グラフの(n-p)-点連結性」の判定を、補題２を利用し「補グラフが、$|V(H)|\geq p+1$である狭義完全二部グラフHを部分グラフとして含まない」という等価な性質を判定することによって、行っている。そのため、補グラフを作成して、その上で種々の操作が行われている。故にこれらのアルゴリズムは、与えられたグラフGの補グラフG^cについて、$\kappa(G^c)\geq n-p$であるか否かの判定や、$\kappa(G^c)$の計算問題に容易に応用できる。よって次の定理を得る。

定理4　与えられた無向単純グラフG=(V,E)及び整数$1\leq p\leq n$に対して、$\kappa(G^c)<n-p$であるならば"no"を、$\kappa(G^c)\geq n-p$であるならば$\kappa(G^c)$の値を出力することは、$O(m+np^{3.5})$時間でできる。また$\kappa(G^c)$の計算は$O(m+n(n-\kappa(G^c))^{3.5})$時間でできる。□

証明は定理２、３と同様であるので省略する。

6. まとめ

本稿では、グラフの点連結度の計算問題について、特に連結度が高い場合に有効なアルゴリズムを検討し、与えられたグラフGが(n-p)-点連結であることを$O(m+np^{3.5})$時間で判定するアルゴリズムを与えた。本アルゴリズムは$\kappa(G)\geq n-p$であるならば$\kappa(G)$の値も同時に計算することができる。また、補グラフG^cについても、同じ計算時間で(n-p)-点連結であることを判定し、$\kappa(G^c)\geq n-p$であるならば$\kappa(G^c)$の値を出力する。

本アルゴリズムは特に定数pに対しては線形時間で動き、既存の最高速のアルゴリズムを使った場合の$O(n^{4.5})$時間に比べて圧倒的に高速である。また、$0<c<1$である任意の定数cに対する$\lfloor cn \rfloor$-点連結性の判定は、本アルゴリズムはこれまでの最高速のアルゴリズムと同じ$O(n^{4.5})$時間を達成している。

また、$\kappa(G)$を計算する$O(m+n(n-\kappa(G))^{3.5})$時間のアルゴリズムを与えた。これを応用すれば$\kappa(G^c)$を$O(m+n(n-\kappa(G^c))^{3.5})$時間で計算することもできる。

通信網の重要性が増すにつれ、高連結度の問題も重要性を増すと考えられるが、本稿で扱った問題の実用性等も含めた検討、アルゴリズムの改善等を今後検討していく。また、補グラフ上のアルゴリズムとの関連も重要な検討課題である。

謝辞　貴重な御意見を頂いた京都大学工学部の茨木俊秀教授ならびに永持仁助教授そして豊橋技術科学大学の増山繁助教授に感謝致します。

参考文献

[1] 永持仁, "グラフの最小カットについて," 藤重悟編, "離散構造とアルゴリズムII," 第４章, 近代科学社 (1993), pp 155-208.

[2] S. Even and R. E. Tarjan , "Network flow and testing graph connectivity," SIAM J. Comput, Vol. 4, No. 4 (1975), pp. 507-518.

[3] S. Even, "An algorithm for determining whether the connectivity of a graph is at least k," SIAM J. Computing, Vol. 4 (1975), pp. 393-396.

[4] Z. Galil, "Finding the vertex connectivity of graphs," SIAM J. Comput., Vol. 9, No. 1 (1980), pp. 197-199.

[5] Nagamochi H. and Ibaraki T., "A linear-time algorithm for finding a sparse k-connected spanning subgraph of a k-connected graph," Algorithmica, No. 7 (1992), pp. 583-596.

[6] D. W. Matula, "Determining edge connectivity in O(nm)," Proc. of 28th IEEE Symp. Foud. COMP. SCI. (1987), pp. 249-251.

[7] M. Y. Kao and S. H. Teng, "Simple and efficient graph compression schemes for dense and complement graphs," Proc. of ISSAC'94, LNCS #834 (1994), pp. 451-459.

[8] 伊藤大雄, "補グラフ入力に対する線形時間グラフ探索アルゴリズム," 信学技報, COMP95-12 (1995), pp. 9-16.