アルゴリズム 66－2
（1999．1．27）

# 有向グラフ上の高枝連結節点部分集合生成アルゴリズム

穂浪 昭二* 伊藤 大雄† 上原 秀幸 横山光雄
豊橋技術科学大学 情報工学専攻

概要

従来の連結度の概念を拡張するものとしてグラフにその節点部分集合(領域)族を与えたものとして定義される領域グラフ上の連結度であるNA(Node-to-Area)連結度が提案されている．本研究では有向グラフにおけるNA枝連結度の性質についての考察を行ない，有向グラフである領域グラフが$k$-NA枝連結である必要十分条件を示している．また，与えられた領域グラフが$k$-NA枝連結でない場合に領域に最小数の節点を付与することで領域グラフを$k$-NA枝連結とする有向$k$-NA枝連結化問題についての考察も行ない，$k=1,2,3$の場合についての多項式時間アルゴリズムを提案している．

# An algorithm for finding a node-subset having high connectivity from other nodes

HONAMI Shoji*, ITO Hiro†, UEHARA Hideyuki, and YOKOYAMA Mitsuo
Department of Informaiton and Computer Sciences,
Toyohasi University of Technology

Abstract

The idea of area graph which consists of a graph and a set of areas (node subset) had been proposed. For considering NA (node-to-area) edge connectivity which is the connectivity between a node and an area on directed graphs, we introduce the idea of directed area graph. In this study, we consider a problem of finding a minimum node subset which makes area graph $k$-NA-edge-connected. This paper gives a nesessary and sufficient condition for $k$-NA-edge connection, and presents polynomial time algorithm for this problem for $k = 1, 2, 3$.

## 1 はじめに

通信網の特性を理論的に評価する際に，交換局を節点に通信リンクを枝にそれぞれ対応させてモデル化したグラフの特性を評価する手法が取られており，通信網の信頼性を評価する際にはモデル化したグラフの連結度が重要な指針となっている．連結度はグラフ理論において古くから研究されてきた概念であるが，これまでの成果の多くは2節点間の連結度を基礎としたものであり，想定しているのは一対一の通信であった．

しかしある特定の交換局の集合が通信網全体に同一のサービスを提供する通信網などでは一対多通信が基本となってくる．このような通信網を理論的に評価するのに適したモデルとして領域グラフが提案されている[1],[2]．領域グラフではサービスを提供する交換局の集合に対応する節点部分集合を領域として定義できるので，領域グラフによりモデル化される通信網の信頼性の指針として節

*honami@yilab.tutics.tut.ac.jp
†ito@tutics.tut.ac.jp

点の領域の間の連結度であるNA連結度 (node-to-areaconnectivity) を用いることが出来る．

本研究では有向領域グラフが $k$-NA枝連結であるための必要十分条件を示している．また，与えられた領域グラフが $k$-NA枝連結でない場合に領域に最小数の節点を付加することで領域グラフを $k$-NA枝連結にする有向 $k$-NA枝連結化問題についても考察を行ない，$k=1,2,3$ についての多項式時間アルゴリズムを提案している．

無向グラフにおける $k$-NA枝連結化問題については $k$ を変数としても多項式時間アルゴリズムが存在し，特に $k=1,2,3$ に対しては線形時間アルゴリズムが存在することが示されている [2].

# 2 準備

本研究では有向グラフを取り扱うのでグラフや連結度などの用語も有向グラフ上の概念として扱う．

## 2.1 諸定義

$G=(V,E)$ を有向多重グラフとする．$V,E$ はそれぞれ節点集合，枝集合であり，グラフ $G$ を明示したい場合にはそれぞれ $V(G),E(G)$ と表す．2つの節点部分集合 $X,Y\,(X\cap Y=\phi)$ に対し，$E(X,Y)$ を $X$ から $Y$ へ向かう枝の集合，すなわち $E(X,Y)=\{(x,y)\in E\,|\,x\in X,\,y\in Y\}$ とする．一要素のみからなる集合 $\{x\}$ は単に $x$ と表す場合がある．隣接している2節点 $x,y$ の間の枝で尾が $x$，頭が $y$ であるものを $(x,y)$ のように表す．

空でない節点真部分集合 $W\subset V$ に対し $E(W,V-W)$ をカットと呼ぶ．カット $E(W,V-W)$ は単に $W$ と表す場合がある．$v\in W$ である節点 $v$ はカット $W$ に属するという．カット $W$ の大きさは $|E(W,V-W)|$ で定義され，大きさ $k$ のカットは $k$ カットと呼ばれる．2つの節点部分集合 $U_1,U_2$ に対して $U_1\subseteq W$ かつ $U_2\subseteq V-W$ であるとき，カット $W$ は $U_1$ から $U_2$ を分離するという．

2つのカット $E(X,V-X),E(Y,V-Y)$ は $X\cap Y\neq\phi$ であるとき重なっているといい，$X\cap Y$ を重なりと呼ぶ．$c$ 個のカットの集合 $\mathcal{W}=\{W_1,W_2,\ldots,W_c\}$ 内の任意のカット $W_i$ について $j\neq i$ であるカット $W_j$ が存在し $W_i\cap W_j\neq\phi$ が成立するとき，$\mathcal{W}$ を重複集合と呼ぶ．$\mathcal{W}$ 内の全てのカットに属する節点部分集合を $\mathcal{W}$ の真の重なりと呼ぶ．

$k$ カットは任意の $X\subset W$ について $|E(W-X,V-(W-X))|>k$ であるとき極小 $k$ カットと呼ばれる．一般的な極小性の定義とは異なり，極小 $k$ カットを含まないような $k$ カットが存在するが，そのようなカット内には $h<k$ である極小 $h$ カットが必ず存在する (図1).

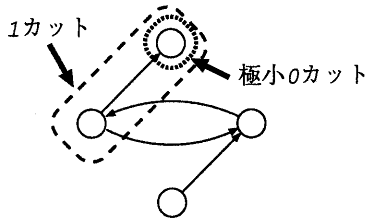



図1: 極小1カットを含まない1カット

グラフ $G=(V,E)$ と節点部分集合族 $\mathcal{X}=\{V_1,V_2,..,V_p\}$ $(V_i\subseteq V)$ の組合せを領域グラフと呼び $(G,\mathcal{X})$ と表す．$\mathcal{X}$ の各要素 $V_i$ を領域と呼ぶ．

$(G,\mathcal{X})$ において $x\in W$ から $V_p\subseteq V-W$ を分離するような大きさ $k-1$ 以下のカット $W$ が存在しないとき，$x$ と $V_p$ は $k$-NA枝連結 ($k$-NA-edge-connected) であるという．また，$x$ と $V_p$ が $k$-NA枝連結である最大の $k$ を $x$ と $V_p$ のNA枝連結度 (NA-edge-connectivity) と呼び $\lambda(v,V_p;G)$ と表す．領域グラフ $(G,\mathcal{X})$ において，任意の領域 $V_i\in\mathcal{X}$ と $V_i$ に含まれない任意の節点 $x\in V-V_i$ に対して $x$ と $V_i$ が $k$-NA枝連結であるならば $(G,\mathcal{X})$ は $k$-NA枝連結であるという．$(G,\mathcal{X})$ が $k$-NA枝連結であるような最大の $k$ を $(G,\mathcal{X})$ のNA枝連結度と呼び $\lambda(G,\mathcal{X})$ と表す。

本研究では次の問題の考察を行なった．この問題では $k$ は定数であり，今回は $k=1,2,3$ の場合の解法を提案している．NA枝連結性は領域ごとに独立した性質であるので，領域グラフに複数の領域が存在する場合にはそれぞれの領域について議論しなければならない．よって本研究では簡単のために領域を $V_1$ 1つのみとし，以下では領域グラフを $(G,V_1)$ と表すものとする．

有向 $k$-NA 枝連結化問題

| | |
|---|---|
| 入力 | : $(G=(V,E),V_1)\ V_1 \subseteq V$ |
| 出力 | : $(G=(V,E),V_1')\ V_1' \subseteq V$ |
| 目的関数 | : $\|V_1'\|-\|V_1\| \rightarrow \min$ |
| 制約 | : $\lambda(G,V_1') \geq k,\ V_1 \subseteq V_1'$ |

領域グラフ $(G,V_1)$ にそれに接続する枝を持たない節点 (孤立点) が存在する場合に領域グラフが $k$-NA 枝連結となるためには任意の孤立点は $V_1$に含まれなければならない．また，$G$ から孤立点を取り除いた領域グラフ $(G',V_1')$ が $k$-NA 枝連結であるならば取り除いた節点を $V_1$ に含むような領域グラフ $(G,V_1)$ も $k$-NA 枝連結である．よって一般性を失うことなく $G$ には孤立点が存在しないと仮定できるので，$|V|=O(|E|)$ と考えて良い．

## 2.2 領域グラフの縮約

領域グラフ $(G,V_1)$ の節点部分集合 $W$ を1つの節点 $w$ に対応させて新しいグラフを作る操作を $(G,V_1)$ の ($W$ についての) 縮約と呼び，この操作により得られる領域グラフを $(G/W,V_1/W)$ と表す．縮約では $E(W,V-W)$, $E(V-W,W)$ の全ての枝は方向を保ったまま $w$ に接続させ，自己ループとなる枝は削除する．また $V_1 \cap W \neq \phi$ ならば $V_1/W$ を $\{V_1-W\} \cup \{w\}$ と定義する．

節点部分集合 $W$ で誘導される $(G,V_1)$ の部分グラフを $(G(W),V_1(W))$ と表す．

# 3 $k$-NA 枝連結性の必要十分条件

## 3.1 $k$-NA 枝連結である必要十分条件

任意の領域グラフ $(G,V_1)$ は $G$ が $k$ 枝連結であるならば $k$-NA 枝連結であるが，一般にその逆は成立しない．与えられた領域グラフ $(G,V_1)$ が $k$-NA 枝連結であるための必要十分条件を示す．

**定理 3.1** *与えられた領域グラフ $(G,V_1)$ が $k$-NA 枝連結であるための必要十分条件は，$G$ の任意の極小 $l$ カット (ただし $0 \leq l \leq k-1$) が $V_1$ の節点を少なくとも 1 つずつ含むことである．*

**証明**

**必要性)** ある $k$-NA 枝連結な領域グラフ $(G,V_1)$ に領域 $V_1$ の節点を 1 つも含まないような極小 $l$ カット $W \subset V$ (ただし $0 \leq l \leq k-1$) が存在すると仮定する．$(G,V_1)$ が $k$-NA 枝連結であるので $w \in W$ から $V_1 \subset V-W$ を分離する大きさ $k-1$ 以下のカットは存在しないことになるが，これは $W$ が極小 $l$ カットであることに矛盾する．

**十分性)** $(G,V_1)$ が $k$-NA 枝連結でないならば，ある節点 $w \in V-V_1$ から領域 $V_1$ を分離する大きさが $k-1$ 以下のカット $W$ が存在する．つまり $W$ 内に $V_1$ の節点を1つも含まないような極小 $l$ カット $Y \subseteq W$ (ただし $0 \leq l \leq k-1$) が存在することになる． □

## 3.2 1-NA 枝連結化問題の解法

定理 3.1より直感的に得られる 1-NA 枝連結化アルゴリズムは次のアルゴリズム CHECK を $k=1$ としたものである．定義より複数の極小 0 カットが共通部分 (重なり) を持つことはないので，この方法が目的関数を満足する事は明らかである．

**procedure** CHECK( $(G,V_1)$, $k$)
**begin**
1 $V_1'=V_1$;
2 **for** 極小 $k-1$ カット $W \subset V(G)$ **do**
**if** $W$ 内に $V_1'$ に属する節点が存在しない
**then** $W$ 内の任意の節点を $V_1'$ に付与する;
3 $(G,V_1')$ を出力する;
**end**

強連結な有向グラフとは任意の節点間が両方向に 1 枝連結でグラフであり，強連結成分とは極大な強連結部分グラフである．有向グラフの極小 0 カットは出枝数 0 の強連結成分に一致するので (図 2)，グラフを強連結成分に分解することで全ての極小 0 カットを求めることが出来る．有向グラフの強連結成分への分解は $O(|E|)$ 時間で可能であることが知られており [6]，各極小 0 カットについて $V_1$ の節点が含まれるか否かを判定する時間はアルゴリズム全体で $O(|V|)$ である．よって $k=1$ の場合の CHECK は全体で $O(|E|)$ 時間で実行可能である．

$k \geq 2$ の場合でも $h=1,2,\ldots,k$ について CHECK $((G,V_1),h)$ を実行することで $k$-NA 枝連結な領域グラフを得ることが出来る．しかし極小 0 カットとは異なり，$l>1$ である複数の極小 $l$

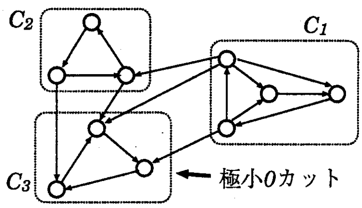



図 2: 有向グラフとその強連結成分

カットでは重なりをもつカットの組が属する集合(重複集合) が存在するので，単純に CHECK を繰り返す方法では目的関数を満足するとは限らない．次節よりこの点を考慮して目的関数を満足するようにした 2,3-NA 枝連結化アルゴリズムについて述べていく．

# 4 2-NA 枝連結化問題

## 4.1 極小 1 カットの重複集合

2-NA 枝連結化問題の目的関数を満足しながら領域グラフを 2-NA 枝連結にするには極小 1 カットの重複集合の性質を明らかにしなければならない．カットの極小性の定義から，重なりを持つ 2 つの極小 1 カット $E(X,V-X) = \{(x,v)\,|\,x \in X, v \in V-X\}$，$E(Y,V-Y) = \{(y,w)\,|\,y \in Y, w \in V-Y\}$ では $x,y \in X \cap Y$，$v \in Y-X$，$w \in X-Y$ が成立する事がいえる (図 3).

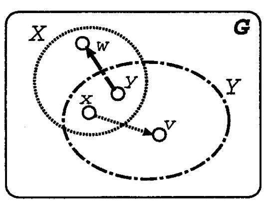



図 3: 重なりを持つ 2 つの極小 1 カット

極小 1 カットのみからなる重複和集合の性質に関するを補題を以下に示す．証明は文献 [3].

**補題 4.1** $\{W_1, W_2, \ldots, W_p\}$ を有向グラフ $G = (V,E)$ 内の極小 *1* カットの集合とする．任意の $q \in \{2,\ldots,p\}$ に対しある $i < q$ が存在し，$W_q \cap W_i \neq \phi$ であるならば $\bigcap_{i=1}^{p} W_i \neq \phi$ が成り立つ．

**補題 4.2** $\{W_1, W_2, \ldots, W_p\}$ を有向グラフ $G = (V,E)$ 内の極小 *1* カットの集合とする．任意の $q \in \{2,\ldots,p\}$ に対しある $j < q$ が存在し $W_q \cap W_i \neq \phi$ であるならば，$\bigcup_{i=1}^{p} W_i$ は極小 *0* カット，または *0* カットを含まないような $V$ になる *(*図 *4)*.

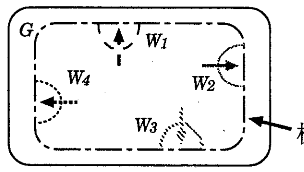



図 4: 複数の極小 1 カットの重複集合

以上を定理としてまとめる．

**定理 4.3** 極小 *1* カットの重複集合は極小 *0* カットを形成し，重複集合内の全ての極小 *1* カットに属する節点部分集合である真の重なりを持つ．

## 4.2 2-NA 枝連結化問題の解法

定理 4.3より，$V_1$ の節点を含まないようなものを 1 つでも含む極小 1 カットの重複集合 $\mathcal{W}$ については，$\mathcal{W}$ の真の重なりから任意に選んだ節点を $V_1$ に付与し，極小 0 カットと重複集合に属さない極小 1 カットのうち $V_1$ の節点を含まないものについてはカットに属する任意の節点を $V_1$ に付与する，という方法で目的関数を満足することができる．実際のアルゴリズムは 6に示す．

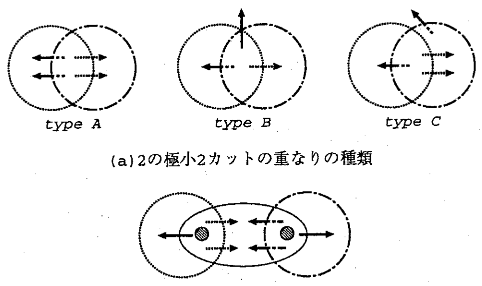
# 5 3-NA 枝連結化問題

## 5.1 極小 2 カットの重複集合

定理 3.1より，領域グラフの 3-NA 枝連結性を議論する際に考慮するのは極小 0 カット，極小 1 カット，極小 2 カットの 3 種類である．極小 0 カットと極小 1 カットについては前節までに述べた方法で最適な領域への節点付与を実行することが可能である．ここでは極小 2 カットの重複集合の性質

を明らかにする事で最適な節点付与の方法を述べていく.

重なりを持つ極小2カットの組は図5 (a) 示すような3種類に分類される. 任意の極小1カットの重複集合には真の重なりが存在したが, 極小2カットの重複集合には真の重なりが存在しない場合がある (図5 (b)).



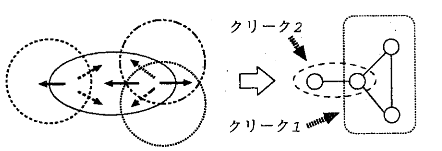

(a) 2の極小2カットの重なりの種類

(b) 3つのカットをカバーするには2点が必要十分な場合

図 5: 極小2カットの重複集合

重複集合内の極小2カットの関係を把握しやすくするために次の性質を満たす補助グラフ $G_r$ を導入した. $G_r$ を用いることで極小2カットのみからなる重複集合に特徴付けを与える補題5.1, 5.2が得られ, それらを一般化することで定理5.3が得られた.

- 極小2カットが節点に対応.
- 共通部分を持つ極小2カット間に枝を付与する. (多重枝は考えない. )




(a) 極小2カットの重なり　　(b) 補助グラフでの形状

図 6: 補助グラフの例

**補題 5.1** 補助グラフ $G_r$ 上でクリークとなるような極小 *2* カットの重複集合は真の重なりを持つ.

**Proof.** 重複集合に属する極小2カットの数 $c$ に関する帰納法で証明を行なう.

$c = 2$ については明らかに成立する.

$k$ 個の極小2カットの重複集合であり, そのうちの任意の $k-1$ 個のカットからなる重複集合では ($k-1$ 個の) 全てのカットに属する節点が存在するようなもの $\mathcal{W}$ について, $\mathcal{W}$ 内の全てのカットに属する節点は存在しないと仮定する. このとき, ある極小2カット $W_k \in \mathcal{W}$ には, $W_k \cap W_i \neq \phi$, $W_k \cap W_j \neq \phi$ かつ $W_k \cap W_i \cap W_j = \phi$ を満たす極小2カット $W_i, W_j \in \mathcal{W}$ が存在する (図7).

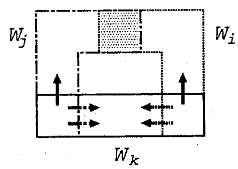



図 7: 仮定により得られる重複集合

$W_k$ が $W_i$ と type A または B で重なっていると仮定すると $W_k$ からの出枝は全て $W_k \cap W_i$ に属する事になる. すると2つの極小2カットの重なり方の定義から $W_k \cap W_j$ が0,1, または2カットとなり $W_k$ が極小2カットであることに矛盾する. よって $W_k$ は $W_i$ と ($W_j$ ともに) type C で重なっていることになる. $W_i, W_j$ はそのような $W_k$ に対して type C で重なっているのでこれらのカットの出枝は全て $W_k \cap W_i$, $W_k \cap W_j$ に属するさなければならない. すると $W_i \cap W_j$ が0カットとなり $W_i, W_j$ が極小2カットである事に矛盾する. □

**補題 5.2** $c$ 個の極小 *2* カットの重複集合の補助グラフでの形状は, 節点数 $c$ のクリークか, *2* つのクリークが *1* つの節点を共有しているもののいずれかになる *(*図 *6(b))*.

**Proof.** 重複集合に属する極小2カットの数 $c$ に関する帰納法で示す.

$c = 2$ については明らかに成立する.

$k$ 個の極小2カットの重複集合であり, そのうちの任意の $k-1$ 個のカットからなる重複集合では補題が成立するようなもの $\mathcal{W}$ について, $\mathcal{W}$ では補題が成立しないと仮定する. つまり $\mathcal{W}$ の全てのカットをカバーするには3つ以上の節点が必要

であると仮定する．定義より任意の重複集合は補助グラフ上では連結な部分グラフとなるので，仮定より $\mathcal{W}$ の部分補助グラフには異なるクリークに属する枝のみからなる長さが 3 以上のパスか T 字型の節点数 4 の木が含まれることになる．しかし，補題 5.1の議論より極小 2 カットの重複集合は補助グラフ上で異なるクリークに属する枝のみからなるパスの長さは 2 となることがいえるので，長さ 3 以上のパスが存在する事はない．また節点数 4 の T 字型の木の内点に対応するカットが存在すると，そのカットが極小 2 カットであることに矛盾が生じる事もいえる． □

補題補題 5.1, 5.2を定理としてまとめる．

**定理 5.3** *真の重なりを持つ極小 2 カットの重複集合は全体で極小 0 カットまたは極小 1 カットとなる．真の重なりを持たない極小 2 カットの重複集合は全体で極小 0 カットとなり，集合内の全てのカットをカバーするには 2 つの節点で必要十分である．*

**Proof.** 真の重なりを持たない重複集合をカバーするには 2 節点で必要十分である事は補題 5.2より明らかである．重複集合全体が極小 0 カットまたは極小 1 カットとなる事の証明は補題 5.4を用いて重複集合に属するカットの数 $c$ に関する帰納法で示すことが出来る．詳細は [4] を参照．

**補題 5.4** *3 個の極小 2 カットからなる重複集合は，真の重なりを持たないならば補助グラフ $G_r$ 上では長さ 2 のパスとなり，集合全体は極小 0 カットとなる (図 5(b) ).*

**Proof.** 補題 5.1の議論から 3 個の極小 2 カットからなる真の重なりを持たない重複集合は補助グラフ $G_r$ 上では長さ 2 のパスとなり，集合全体が 0 カットとなることは明らかである．重複集合内に 0 カットが存在すると，集合に属するあるカットが極小 2 カットである事に矛盾する事から重複集合全体が極小 0 カットである事が示される． □

以上は極小 2 カットのみからなる重複集合の性質である．極小 1 カットと極小 2 カットを同時に含む重複集合が考えられるが，定理 5.3とカットの極小性の定義からそれらはある極小 2 カットの重複集合がある極小 1 カットの部分集合になっている場合に限られることがいえる．

### 5.2 3-NA 枝連結化問題の解法

定理 5.3より，3-NA 枝連結化問題の目的関数を満足するような領域への節点付与の方法は，極小 1 カット，極小 0 カットについては 2-NA 枝連結化問題と同様に選び，極小 2 カットについては，i) 他の極小 2 カットと重なっていなければカット内の任意の節点を選ぶ，ii) 真の重なりを持つ重複集合に属するならば真の重なり内から任意に選ぶ，iii) 真の重なりを持たない重複集合に属するならば 2 つで集合内の全てのカットをカバーするような節点部分集合の組からそれぞれ任意に 1 点ずつ選ぶ，というものになる．よってこれらを効率よく発見するアルゴリズムがあれば良いことになる．

## 6 2,3-NA 枝連結化アルゴリズム

### 6.1 前処理

本研究で提案する 2,3-NA 枝連結化問題を解くアルゴリズム 2,3-DNAEC-A では，極小カットの重複集合を単純に扱うために，入力の領域グラフ $(G, X)$ を部分領域グラフの集合 $\mathcal{G}$ に分割する次の操作が前処理として用いられている (図 8).

**procedure** CONTRACT $(G, V_1)$
  **begin**
1 $\mathcal{G} = \{(G, V_1)\}$;
2 **while** $(G', V_1') \in \mathcal{G}$ に節点数 2 以上の
    0 カット $W$ が存在
   **do** $\mathcal{G} = \{(\mathcal{G} - (G', V_1'))\} \cup$
   $\{(G'/W, V_1'/W)\} \cup \{(G'(W), V_1'(W))\}$;
3 $\mathcal{G}$ を出力;
  **end**

有向グラフの強連結成分分解の構造木 [6] には任意の 0 カットに対応する 0 カットが存在する．つまり構造木を探索することでグラフ中の全ての 0 カットを求められるので，CONTRACT は $O(|E|)$ 時間で実行可能である．CONTRACT で得られる $\mathcal{G}$では以下の補題 6.1, 6.2が成立する．(証明は文献 [3]．)

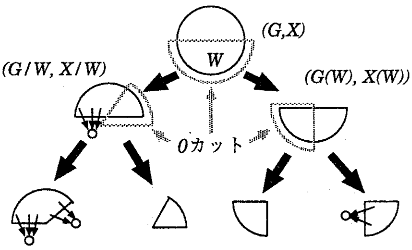



図 8: 前処理の概要

**補題 6.1** 領域グラフ $(G, V_1)$ に *0* カット $W \subset V$ が存在する場合，$W$ を縮約した領域グラフ $(G/W, V_1/W)$ と，$W$ で誘導される部分領域グラフ $(G(W), X(W))$ に対して次の関係が成立する．

$$\lambda(G, V_1) = \min\{\lambda\big(G/W, X/W\big), \lambda\big(G(W), X(W)\big)\}$$

**補題 6.2** *CONTRACT* によって得られた領域グラフは次の *3* 種類に分類できる．

**type 1** 節点数 *1* の *0* カットを含むもの

**type 2** *0* カットを持たず極小 *1* カットに属する節点のみからなるもの．

**type 3** *0* カットを持たず極小 *1* カットの重なりを含まないもの

補題 6.1より，$\mathcal{G}$ では入力領域グラフの NA 枝連結度が保たれる事が保証されている．また補題 6.2より，$\mathcal{G}$ の要素のうち type 1,3 領域グラフでは極小 1 カットの重復集合は存在せず (ただし極小 2 カットの重復集合は極小 1 カット内に存在しうる)，入力の領域グラフの任意の極小 1 カットの重復集合と極小 2 カットの重復集合のうち全体が 0 カットとなるようなものは $\mathcal{G}$ において type 2 の領域グラフとなることがいえる．

## 6.2 2,3-DNAEC-A アルゴリズム

本研究で提案する $k = 2,3$ に対して $k$-NA 枝連結化問題を解くアルゴリズム 2,3-DNAEC-A を以下に示す．定理 3.1より step 2 で $\mathcal{G}$ の全ての領域グラフを $k$-NA 枝連結にしている事は明らかであり，これらの領域グラフを結合して得られる $(G, X')$ が $k$-NA 枝連結であることは補題 6.2より保証される．これよりこのアルゴリズムで最適な領域への節点の付与が実行可能であることを示していく．

**procedure** 2,3-DNAEC-A $\big((G, V_1), k(=2,3)\big)$
**begin**
1 $\mathcal{G} =$ CONTRACT $(G, V_1')$;
2 **for** $(\hat{G}, \hat{V}_1) \in \mathcal{G}$ **do**
  **if** $(\hat{G}, \hat{V}_1)$ が type 2
  **then**
    **for** $1 \le l \le k-1$ **do** CHECK$\big((\hat{G}, \hat{V}_1), l\big)$;
  **else**
    **for** $0 \le l \le k-1$ **do** CHECK$\big((\hat{G}, \hat{V}_1), l\big)$;
3 $\mathcal{G}$ の各領域グラフを結合して
  領域グラフ $(G, V_1')$ を作成し出力する;
**end**

$\mathcal{G}$ の要素である領域グラフ数は $O\big(|V|\big)$ となる．各領域グラフについて極小 $l$ カット (ただし $0 \le l \le k-1$) の列挙と type の判別を行なう必要があるが，このために入力領域グラフ $(G, V_1)$ における任意の節点間の枝連結度が 0,1,2, または 3 以上，のいずれかを判定可能な行列 $M$ を導入する．有向グラフの 2 節点間の枝連結度が 3 以上であるか否かの判定は $O\big(|E|\big)$で可能であり，枝連結度が 3 以下の場合は同時にその枝連結度も得られる事が知られている [5] ので，この行列は $O\big(|V|^2|E|\big)$ 時間で作成可能である．

領域グラフの type は 0 カットの有無で type 1 と 2,3 を判別でき，グラフの全ての節点から極小 1 カットで分離されない節点部分集合の有無で type 2,3 の判別できる．これにはその領域グラフの全ての節点について $M$ を探索するれば良い．領域グラフの全ての節点について $M$を探索して $l$ カットで分離されない節点部分集合を列挙していくことで極小 $l$ カットを列挙することが可能である．よって type の判別に必要な手間はアルゴリズム全体で $O\big(|V|^3\big)$ 時間となる．各領域グラフについてグラフ中の全ての極小 $l$ カット内に $V_1$ の節点が存在するか否かの判定と，存在しなかった場合に行なう $V_1$ への節点の付与には $O\big(|V|\big)$ 時間が必要であるので，アルゴリズム全体では $O\big(|V|^2\big)$ 時間となる．よって step 2 全体の計算時間は $O\big(|V|^2|E|\big)$ 時間となる．

ただし $k=3$ では CHECK を実行する順序に注意する必要がある．極小 1 カットの重複集合の内

に極小2カットの重複集合が存在する場合，極小カットの定義から，極小2カットの重複集合において領域に付与される節点の候補の節点部分集合は極小1カットの重複集合におけるそれの部分集合となる．よって極小1カットに関する節点付与を先に行なうと，極小2カットについても節点付与が行なわれる場合が生じてしまう．よって，$k=3$ では CHECK は $l$ の降順で実行しなければならない．

最後に計算時間を見積もる．CONTRACT の計算時間が $O(|E|)$ 時間であるのでその逆の操作である step 3 も同じ計算時間で実行可能である．$\mathcal{G}$ の各要素 (領域グラフ) の入力グラフにおける包含関係は2分木で表現されるので (図8) 各領域グラフを2分木における後行順に操作していけば，2分木上で子を持つ領域グラフを操作する際には子は既に $k(=2,3)$-NA 枝連結となっているので，この時点で子の領域グラフを親に結合していけば良い．よって，アルゴリズム全体の計算時間は step 2 の計算時間に一致し，$O(|V|^2|E|)$ 時間となる．

## 7 まとめ

本稿では，有向グラフ $G=(V,E)$ に領域 $V_1 \subseteq V$ を与えたものである領域グラフ $(G,V_1)$ における節点と領域の間の枝連結度である NA 枝連結度について，与えられた領域グラフが $k$-NA 枝連結であるための必要十分条件と，$k=1,2,3$ については，$k$-NA 枝連結でない領域グラフを $k$-NA 枝連結とする $k$-NA 枝連結化問題を解く多項式アルゴリズムを示した．ただし，これらの成果は複数の領域を持つ領域グラフ $(G, X=\{V_1,V_2,\ldots,V_p\})$ に対しても応用が可能なものである (表1).

今後の課題として，$k \le 3$ についてのアルゴリズムの高速化や，$k \ge 4$ に対する NA 枝連結化アルゴリズムの導出などが挙げられる．

表 1: アルゴリズムの計算時間

| $k$ | $k$-DNAEC-A | 複数領域の場合 |
|---|---|---|
| 1 | $O(\|E\|+\|V_1'\|)$ | $O(\|V\|+L')$ |
| 2 | $O(\|V\|^2\|E\|)$ | $O(\|V\|^2\|E\|+\|V\|L')$ |
| 3 | $O(\|V\|^2\|E\|)$ | $O(\|V\|^2\|E\|+\|V\|L')$ |

ただし，$L'=\sum_{i=1}^{p}|V_i'|+|V|$

## 参考文献


[1] 伊藤 大雄, “グラフにおける節点・領域連結度について”, 電気学会論文誌 C, Vol. 114-C, No. 4, pp. 463-46, 1994.

[2] H. Ito and M. Yokoyama, “Edge Connectivity between Nodes and Node-Subsets”, Networks, Vol. 31, No. 3, pp.157-164, 1998.

[3] 穂浪 昭二, 伊藤 大雄, 上原 秀幸, 横山 光雄, “有向グラフにおける 1,2-NA 枝連結化問題”, 第33回 SSOR 予稿集, pp.24-29, 1998.

[4] 穂浪 昭二, “有向グラフにおける $k$-NA 枝連結化問題”, 修士論文, 豊橋技術科学大学, Feb., 1998. (to appear)

[5] S. Even and R.E. Tarjan, “Network Flow and Testing Graph Connectivity”, SIAM J. Comput, Vol. 4, No. 4, pp.507-518, 1975.

[6] 浅野 孝夫, “情報の構造 上 データ構造とグラフアルゴリズム”, 日本評論社, 1994.