人工知能学会研究会資料
SIG-FPAI-B502-15

可変接続数下限を用いた孤立性 j-核

Isolated Variable j-Core Based on Vertex-Dependent Connection Lower Bound

趙 奇[1] 原口 誠[1]* 大久保 好章[1] 富田 悦次[2]
Q. ZHAO M. HARAGUCHI Y. OKUBO E. TOMITA

[1] 北海道大学大学院情報科学研究科
Graduate School of Information Science and Technology, Hokkaido University
[2] 電気通信大学先進アルゴリズム研究ステーション
Advanced Algorithms Research Laboratory, The University of Electro-Communications

Abstract: In this paper, we are concerned with a problem of enumerating *maximal isolated cliques* (*MIC*) in a given undirected graph. Each target to be extracted is defined as a maximal subset X of vertices which is a clique and satisfies an isolatedness. Our isolation concept is based on the notion of *variable j-coreness* which imposes a connection lower bound depending on each vertex in X. Based on a standard clique enumerator, we design a depth-first algorithm for the problem. In our algorithm, we can prune many hopeless cliques from which we can never obtain our solution. It is noted that our solution MICs can be divided into two classes, *fixpoint solutions* and *non-fixpoint solutions*, where the latter ones are not trivial to detect. Based on a degree descending order of vertices, we show a theoretical property of non-fixpoint MICs and present an effective method of detecting them.

1 はじめに

所与のネットワークからの，孤立性の高い密な部分グラフの抽出に向けて，本稿では，極大な孤立性の高いクリークの全列挙問題について考察する．

ネットワークからのコミュニティ抽出においては，現実的なコミュニティ構造のモデルとして疑似クリークを考えることが自然である [1]．これまで，様々な疑似クリークの列挙法が提案され，それらを用いたコミュニティ抽出が議論されてきたが，疑似クリークの総数は一般に膨大となり，とりわけ，オーバーラップする多数の疑似クリーク群が観測されることから，結果としてコミュニティの境界が不鮮明なものとなる．こうした問題を解決するためには，境界が鮮明な疑似クリーク，すなわち，外部との接続性を考慮した孤立性の高い疑似クリークのモデル ([5]) が不可欠である．その足掛かりを得るために，本稿では，孤立性の高いクリーク抽出問題について議論する．

孤立性を考慮したクリークのモデルとして，c-**孤立クリーク**が提案されている [4]．そこでの抽出対象は，極大クリークの中で孤立性の高いものに限定されるが，言うまでもなく，極大でないクリークの中にも，孤立性の高いものが存在することから，多数の孤立クリークが見落とされていることが想像できる．また，その孤立性の定義は，コミュニティ全体としての外部接続数を考慮したものであり，個々の頂点の孤立性は考慮されない．結果として，外部との接続度合いの高い少数の頂点を含むクリークが抽出されてしまう．

本稿で考察する**極大孤立性クリーク**は，こうした問題点を解消したモデルと位置付けることができる．極大孤立性クリークは，孤立性を有するクリークの中で極大なものを指し，文献 [4] の枠組みでは見落とす多くの孤立クリークを救済できる．また，本稿での孤立性は，コミュニティを構成する各頂点に依存した内部接続数下限を要請する**可変** j-**核性**に基づくものであり，適切な内部接続数下限関数を定めることで，個々の頂点の孤立性を制御できる．

本稿では，所与のグラフ G における極大孤立性クリークの全列挙問題に対するアルゴリズムについて考察する．詳細については本論で述べるが，極大孤立性クリークは G のある極大クリークの可変 j-核として出現する．よって，素朴な列挙法として，G の極大クリーク C を列挙し，その可変 j-核を求めることが考

*連絡先：原口 誠
〒 060-0814 札幌市北区北 14 条西 9 丁目
北海道大学大学院情報科学研究科
E-mail: mh@ist.hokudai.ac.jp

えられる．極大クリーク列挙については，文献 [2] に代表される実用上十分高速なアルゴリズムが提案されており，また，可変 j-核演算も，$|C|$ と最大次数の積のオーダで実行できる．しかし，解となる極大孤立性クリークを包含する極大クリークは一般に複数存在することから，解の重複枚挙の回避が不可欠となり，その処理は単純ではない．本稿では，標準的な深さ優先クリーク探索処理に，極大孤立性クリークが有する性質に基づく枝刈り機構と解の判定機構を組み込むことで，極大孤立性クリークを重複なく高速に全列挙する手法を提案する．

2 準備

V を頂点集合，$\Gamma(v)$ を頂点 $v \in V$ の隣接頂点集合とする無向グラフを $G = (V, \Gamma)$ と表す．特に，$\Gamma(v)$ は v 自身を含まないとする．$|\Gamma(v)|$ を頂点 v の次数と呼び，$deg(v)$ で参照する．頂点集合 $X \subseteq V$ と頂点 $v \in X$ について，$\Gamma(v) \cap X$ を $\Gamma_X(v)$ と表し，$|\Gamma_X(v)|$ を $deg_X(v)$ で参照する．

頂点集合 $X \subseteq V$ について，X に誘導される G の部分グラフを $G[X]$ と表す．特に，$G[X]$ における任意の異なる頂点のペアが隣接する時，$G[X]$ を G のクリークと呼ぶ．表記を簡略化するため，クリーク $G[X]$ を，それを誘導する頂点集合 X により参照する．

クリーク $X \subseteq V$ と頂点 $v \in (V \setminus X)$ について，$X \cup \{v\}$ がクリークである時，v を X の候補頂点と呼び，X の候補頂点の集合を $Cand(X)$ と表す．なお，以下では，頂点集合 $X \cup \{v\}$ を Xv と略記する．定義より，$Cand(X) = \bigcap_{v \in X} \Gamma(v)$ であり，特に，$X \subseteq Y$ なるクリーク X と Y について，$Cand(X) \supseteq Cand(Y)$ となる．

3 可変 j-核と孤立性 j_τ-核

3.1 可変 j-核

所与のグラフ $G = (V, \Gamma)$ の頂点集合 $X \subseteq V$ について，各頂点 $v \in X$ の $G[X]$ における次数が定数 j 以上である，すなわち，$deg_X(v) \geq j$ を満たす時，X は *(***定数***)* j-**核性**を有するという [3].

定義より，定数 j-核性を有する頂点集合 X と Y について，$X \cup Y$ もまた定数 j-核性を有する．すなわち，任意の頂点集合 $X \subseteq V$ について，定数 j-核性を有する X の最大部分集合が存在し，これを $G[X]$ の**定数** j-**核**と呼ぶ [3].

本稿では，これを拡張した**可変** j-**核性**の概念を新たに導入する．

いま，頂点集合を定義域，非負の実数を値域とする関数 $j : V \rightarrow \mathbb{R}_{\geq 0}$ を考える．頂点集合 $X \subseteq V$ について，各頂点 $x \in X$ が $deg_X(x) \geq j(x)$ を満たす時，X は**可変** j-**核性**を有するという．定数 j-核性とは異なり，可変 j-核性は，部分グラフ内部の接続数下限を頂点毎に定めたものであり，より柔軟に密な部分グラフを扱うことができる．

定数 j-核性と同様，可変 j-核性を有する頂点集合 X と Y について，$X \cup Y$ もまた可変 j-核性を有することは容易にわかる．よって，任意の頂点集合 $X \subseteq V$ について，可変 j-核性を有する X の最大部分集合が存在し，これを $G[X]$ の**可変** j-**核**と呼ぶ．以下では，$G[X]$ の可変 j-核を $core_j(X)$ で参照する．

定数 j-核同様，可変 j-核についても次の単調性が成り立つ．

観察 1 (可変 j-核演算の単調性) $X_1 \subseteq X_2$ なる任意の頂点集合 $X_1, X_2 \subseteq V$ について，$core_j(X_1) \subseteq core_j(X_2)$ である． ■

頂点集合 X の 可変 j-核を求める手続きは単純である．具体的には，$G[X]$ において次数が $j(x)$ 未満の頂点 x を X から順次除去する操作を繰り返す．一般に，ある頂点の除去は，その他の頂点次数を減少させることから，こうした頂点除去操作を，除去可能な頂点がなくなるまで繰り返せばよい．

3.2 孤立性を考慮した可変 j_τ-核

可変 j-核においては，関数 j により頂点毎に部分グラフ内部の接続数下限を与えるが，これは，外部への接続数上限を暗に考慮していることを意味する．内部接続数の増加に伴い，外部接続数は必然的に減少することから，本稿では，頂点次数のうち，内部接続に用いる割合の下限を定めるパラメータ τ $(0 < \tau \leq 1.0)$ を用いて，任意の頂点 $x \in V$ について，

$$j_\tau(x) = \tau \cdot deg(x) \tag{1}$$

なる関数 j_τ を定義する．頂点 v の外部接続数の上限値は $(1.0 - \tau) \cdot deg_G(v)$ で与えられることから，τ を大きくすることで，外部との接続が殆ど無い可変 j_τ-核が得られることになる．特に，$\tau = 1.0$ の場合は，外部接続数が 0 となり，すなわち，完全孤立状態の可変 j_τ-核となる．この様に，τ は j_τ-核の孤立性を制御することから，ここではそれを**孤立性パラメータ**と呼ぶ．

なお，j_τ-核演算においても，観察 1 で述べた単調性が成り立つことは言うまでもない．

頂点集合 X の 可変 j_τ-核を求める手続きは，先の可変 j-核と同様であり，$G[X]$ において次数が $j_\tau(x)$ 未満の頂点 x を X から順次除去する操作を繰り返せ

```
procedure JCore(G, X, τ):
  begin
    Create vert  with |X|-elements;
    head = 0; tail = |X| − 1;
    for each v ∈ X do
      if τ ≤ deg_X(v)/deg(v) then // v is a core candidiate
        vert[tail] = v; d(v) = deg_X(v); index(v) = tail;
        tail--;
      else // v must be removed
        vert[head] = v; d(v) = deg_X(v); index(v) = head;
        head++;
      endif
    endfor
    head = 0;
    while head ≤ tail do
      v = vert[head];
      for each w ∈ Γ_X(v) do
        if tail < index(w) then // w is a core candidate
          d(w)--;
          if τ < d(w)/deg(w) then // w must be removed
            tail++;
            if tail ≠ index(w) then
              vert[index(w)] = vert[tail]; vert[tail] = w;
            endif
          endif
        endif
      endfor
      head++;
    endwhile
    while head ≤ |X| − 1 do
      print vert[head]; head++;
    endwhile
  end
```

図 1: 可変 j_τ-核抽出アルゴリズム

ばよい．可変 j_τ-核抽出アルゴリズムの疑似コードを図 1 に示す．アルゴリズム中，`vert` は頂点の配列であり，頂点 v が格納された `vert` におけるインデックスは $index(v)$ で参照できるものとし，また，頂点 v の次数情報も $d(v)$ で参照できるものとする．

主要な処理は，後半の `while` ループであり，そこでは，X 中の各頂点について，その隣接頂点をチェックする．よって，グラフ G の最大次数を D とすると，計算量は $|X| \cdot D$ のオーダとなる．

4　極大孤立性クリーク

前節で議論した孤立性を考慮した可変 j_τ-核に基づき，ここでは，本稿における抽出ターゲットとなる**極大孤立性クリーク**について述べ，その列挙問題を定義する．

定義 1 (極大孤立性クリーク)
$G = (V, \Gamma)$ をグラフ，τ を孤立性パラメータとする．G のクリーク $X \subseteq V$ が次を満たす時，X を**孤立性クリーク**と呼ぶ：X は j_τ 核性を満たす，すなわち，任意の頂点 $v \in X$ について，$|X| > \tau \cdot deg(v)$ である．

特に，X を真に包含する G の孤立性クリークが存在しない時，X は極大であると言い，これを**極大孤立性クリーク**と呼ぶ．■

定義 2 (極大孤立性クリーク列挙問題)
所与のグラフ $G = (V, \Gamma)$ と孤立性パラメータ τ について，G の極大孤立性クリークをすべて抽出する問題を極大孤立性クリーク列挙問題と呼ぶ．■

G の極大クリークは，必ずしも j_τ-核性を満たす頂点集合とはならないことに注意する．一般に，極大孤立性クリークは，ある極大クリークの可変 j_τ-核として出現する．よって，それらの素朴な列挙法として，G の極大クリーク C を列挙し，その可変 j_τ-核，すなわち，$core_{j_\tau}(C)$ を求めることが考えられる．極大クリーク列挙については，文献 [2] に代表される実用上十分高速なアルゴリズムが提案されており，また，$core_{j_\tau}(C)$ の演算も，$|C|$ と最大次数の積のオーダで実行できることから，こうした素朴なアプローチも然程悪くない様に思える．しかし，解となる極大孤立性クリークを包含する極大クリークは一般に複数存在することから，解の重複枚挙の回避が不可欠となり，その処理は単純ではない．以下では，標準的な深さ優先クリーク探索処理に，極大孤立性クリークが有する性質に基づく枝刈り機構を組み込むことで，極大孤立性クリークを重複なく高速に全列挙する手法を与える．

5　極大孤立性クリークの全列挙

本稿で提案する極大孤立性クリークの全列挙法は，標準的な深さ優先クリーク探索を基礎とする．探索過程で得られるクリークが，後に議論する性質を満たした段階で，それが解として出力すべき極大孤立性クリークか否かの判定を行う．また，クリーク探索過程において，解に到達し得ないクリークが生成された場合は，直ちにバックトラックすることで，無駄な探索枝の展開を抑制する．

5.1　深さ優先クリーク探索

所与のグラフ $G = (V, \Gamma)$ の任意のクリークは，V に関する集合列挙木を辿ることで，重複なく，かつ，完全に列挙することができる．

いま，V 上の全順序 $\prec$ を仮定し，任意の頂点集合は $\prec$ に従いソーティングされているとする．ここで，頂点集合 X の最後尾の頂点を $tail(X)$ で参照する．

クリーク X は，$Cand(X)$ 中の候補頂点 v を用いて，より大きなクリーク Xv へと拡張できる．クリークは逆単調性を有する，すなわち，クリークの任意の部分集合もまたクリークであるから，初期クリーク $X = \emptyset$ $(Cand(X) = V)$ から，こうした拡張処理を繰返すことで，すべてのクリークを生成することができる．その際，X の拡張処理に用いる頂点 $v \in Cand(X)$ を

$tail(X) \prec v$ なるものに限定することで，重複のないクリーク列挙が実現される．

上記処理は，クリーク探索に関する標準的な手法となっており，特に，メモリ効率の観点から，実際の探索過程では深さ優先戦略が採用される．

5.2 有望なクリーク

極大孤立性クリークの全列挙手法の詳細を議論するにあたり，可変 j_τ-核性に関するいくつかの重要な性質を明らかにしておく．なお，以下の議論では，クリーク X とその候補頂点集合 $Cand(X)$ について，$core_{j_\tau}(X \cup Cand(X))$ を $ext(X)$ と表記する．

観察 2 クリーク X と，$X \subseteq Y$ なる孤立性クリーク Y について，

$$X \subseteq Y \subseteq ext(X) = core_{j_\tau}(X \cup cand(X))$$

が成り立つ．

証明： Y は $X \subseteq Y$ なるクリークであるから，$X \subseteq Y \subseteq X \cup cand(X)$ なる関係がある．いま，可変 j_τ-核演算の単調性から，$core_{j_\tau}(Y) \subseteq core_{j_\tau}(X \cup cand(X))$ であり，特に，Y は可変 j_τ-核性を有することから，$Y = core_{j_\tau}(Y)$ となる．よって，$X \subseteq Y \subseteq ext(X) = core_{j_\tau}(X \cup cand(X))$ が示される． ■

上記性質は，クリーク X の拡張処理によって解を得るための必要条件件を与える．$X \subseteq ext(X)$ の場合，X を拡張することで解に到達できる可能性がある．一方，$X \not\subseteq ext(X)$ の場合，X の拡張処理によって解が得られる可能性はないことから，X の拡張処理を直ちに，かつ，安全に打ち切ることができる．以下では，前者の場合の X を**有望なクリーク**と呼ぶ．

5.3 不動点解としての極大孤立性クリーク

観察 2 より，(有望な) クリーク X を真に包含する孤立性クリーク Y が存在する場合は，X は $ext(X)$ の真部分集合，すなわち，$X \subset ext(X)$ であることが容易にわかる．逆に，$X = ext(X) = core_{j_\tau}(X \cup cand(X))$ の場合，X は可変 j_τ-核性を満たし，かつ，それを真に包含する孤立性クリークが存在しないことを意味する．よって，X それ自身が極大な孤立性クリークとなり，解として出力すべきものであることがわかる．

観察 3 クリーク X について，$X = ext(X)$ が成り立つ時，X は極大孤立性クリークであり，特にこれを**不動点解**と呼ぶ． ■

観察 2 より，有望なクリーク $X(\subseteq ext(X))$ について，極大孤立性クリーク Y は必ず $ext(X)$ の部分集合として出現する．よって，X の拡張処理は，$ext(X) \setminus X$ 中の頂点のみを用いて行えば十分である．以下では，$ext(X) \setminus X$ を $K(X)$ と表記し，これを X に関する **K-候補集合**と呼ぶ．

X が不動点解 $(X = ext(X))$ の時，かつ，その時に限り，$K(X) = \emptyset$ となることは容易にわかる．

観察 4 $K(X) = \emptyset$ なる有望なクリーク X は極大孤立性クリークである． ■

一方，この逆は必ずしも真ではなく，X が極大孤立性クリークであっても，$K(X) = \emptyset$ になるとは限らない．つまり，不動点解以外の解を漏れ無く抽出するためには，以下で議論する $K(X) \neq \emptyset$ なる X の処理が重要となる．

5.4 偽 K-候補を伴う極大孤立性クリーク

有望なクリーク X について，$K(X) \neq \emptyset$ なる場合を考える．$K(X)$ が空でないことから，X を真に包含する孤立性クリークが存在する様に思えるが，こうした場合でも，X が極大孤立性クリークとなる可能性がある．すなわち，$K(X)$ 中の頂点は見かけだけの**偽 K-候補**であり，実際の拡張処理には何ら寄与しない．こうした偽候補を検出し，不動点解ではない解 X を抽出するためには，$K(X)$ 中の任意のクリーク C について，$X \cup C$ が可変 j_τ-核性を満たさないことを確かめれば十分であるが，その計算負荷は無視できない．そこで，本稿では，頂点間の順序を工夫することで，偽候補を伴う極大孤立性クリークの性質を明らかにし，それに基づく偽候補の検出法を与える．

いま，所与のグラフ G の頂点集合 V 上に，次数降順の全順序 $\prec$ を導入する [1]．

$$\text{任意の } u, v \in V \text{ について}, u \prec v \Leftrightarrow deg(u) \geq deg(v).$$

以下では，任意の頂点集合は $\prec$ に従ってソーティングされていると仮定する．

有望なクリーク X の拡張には K-候補集合 $K(X)$ 中の頂点を用いることは先に述べたが，特に，$tail(X) \prec x$ なる頂点のみを考えればよい．つまり，こうした頂点の集合を $NL(X)$ とすると，

$$NL(X) = \{x \in K(X) \mid tail(X) \prec x\}$$

となる．

ここで，偽候補を伴う極大孤立性クリークについて，次の性質が成り立つ．

[1] 次数が同じ頂点間には適当な順序を与える．

観察 5 $K(X) \neq \emptyset$ なる極大孤立クリーク X について，$NL(X) = \emptyset$ である． ■

証明： $NL(X) \neq \emptyset$ と仮定する．X は可変 j_τ-核性を有するクリークであるから，任意の頂点 $x \in X$ について，$deg_X(x) = |X| - 1 \geq \tau \cdot deg(x)$ である．ここで，頂点順序 $\prec$ より，任意の頂点 $v \in NL(X)$ について，$deg(x) \geq deg(v)$ が成り立つ．また，$NL(X) \subseteq Cand(X)$ より，v は X のすべての頂点と隣接するから，$deg_{Xv}(v) = |X| > |X| - 1 \geq \tau \cdot deg(v)$ を得る．これより，Xv は可変 j_τ-核性を有するクリークとなるが，このことは X の極大性と矛盾する．よって，$NL(X) = \emptyset$ であることが示される． ■

観察 5 より，探索過程における有望なクリーク X について，$K(X) \neq \emptyset$，かつ，$NL(X) = \emptyset$ である場合のみ，X が極大孤立性クリークか否かを調べればよいことがわかる．その際，素朴には，部分グラフ $G[K(X)]$ の任意の (非空な) クリーク C について，$X \cup C$ の可変 j_τ-核性を調べる必要があるが，実際にはそれをせずに済ますことが可能である．

5.5 偽 K-候補の検出処理

有望なクリーク X について，$K(X) \neq \emptyset$，かつ，$NL(X) = \emptyset$ であると仮定する．いま，$\emptyset \subset C \subseteq K(X)$ なるクリーク C で，$X \cup C$ が可変 j_τ-核性を満たすものが存在するか否かを調べたい．この時，次の性質が成り立つ．

観察 6 $C \subseteq K(X)$ なる $X \cup C$ が可変 j_τ-核性を有するために必要な各頂点 $x \in C$ の $G[K(X)]$ における最小次数は

$$\max\{0, \tau \cdot deg(x) - |X|\}$$

である．

証明： $C \subseteq K(X)$ なる頂点集合 $X \cup C$ が可変 j_τ-核性を有するためには，頂点 $x \in C$ の $G[X \cup C]$ における次数 $deg_{X \cup C}(x)$ は，少なくとも $\tau \cdot deg(x)$ 必要となる．いま，Xx はクリークであるから，$G[C]$ における x の次数 $deg_C(x)$ は少なくとも $\tau \cdot deg(x) - |X|$ であることがわかる．よって，$C \subseteq K(X)$ より，$G[K(X)]$ における x の次数もまた，少なくとも $\tau \cdot deg(x) - |X|$ となる．ここで，$\tau \cdot deg(x) - |X|$ は負の値をとることもあるが，頂点次数は非負であるから，これは $\max\{0, \tau \cdot deg(x) - |X|\}$ と表せる． ■

以下，各頂点 $x \in K(X)$ について，

$$j_{K(X)}(x) = max\{0, \tau \cdot deg_G(x) - |X|\} \qquad (2)$$

で定義される関数 $j_{K(X)} : K(X) \to \mathbb{R}_{\geq 0}$ を考える．

頂点 $x \in K(X)$ について，$j_{K(X)}(x)$ は，x を含む $K(X)$ の部分集合であるクリーク C と X の和集合が可変 j_τ-核性を有するために必要な $G[K(X)]$ における最小次数を与える．よって，こうした可変 j_τ-核性を有する $X \cup C$ が存在するならば，C は $G[K(X)]$ の極大クリーク Q_i の部分集合であり，特に，Q_i の可変 $j_{K(X)}$-核の部分集合となる．

ここでの主目的は，X が偽候補を伴う極大孤立性クリークであるか否かを確かめることであるから，可変 j_τ-核性を有する X を考えれば十分である．この時，Q_i の可変 $j_{K(X)}$-核，すなわち，$core_{j_{K(X)}}(Q_i)$ について，$core_{j_{K(X)}}(Q_i) \neq \emptyset$ の場合は，$X \cup core_{j_{K(X)}}(Q_i)$ は j_τ-核性を有するクリークとなり，X が極大孤立性クリーク (解) ではないことがわかる．

逆に，$G[K(X)]$ の任意の極大クリーク Q_i について，$core_{j_{K(X)}}(Q_i) = \emptyset$ ならば，X は抽出すべき極大孤立性クリークとなり，結果として $K(X)$ 中の頂点が偽候補であったことが判明する．

上記処理においては，$G[K(X)]$ の極大クリークを列挙する必要があるが，その負荷はそれほど高くならないことが期待できる．なぜなら，$K(X) \subseteq cand(X)$ より，$K(X)$ 中の頂点 v は，任意の $x \in X$ と隣接する必要があり，こうした頂点 v は高々$min_{x \in X}\{deg(x)\} - |X| + 1$ しか存在しない．また，$K(X) \neq \emptyset$，かつ，$NL(X) = \emptyset$ となるのは，X のサイズがある程度大きくなってからであると予想されることから，実際の $K(X)$ のサイズは然程大きくはならず，効率的な極大クリーク列挙器を用いることで，高速に処理可能であると期待できる．

さらに，実際は，極大クリークに至る途中段階のクリーク C において，$core_{j_{K(X)}}(C) \neq \emptyset$ になり次第，X が解ではないと判定できることから，こうした処理を組み込むことで，X が解であるか否かの判定をさらに高速化できることを付け加えておく．

6 極大孤立性クリークの全列挙アルゴリズム

これまでの議論に基づく極大孤立性クリークの全列挙アルゴリズムの疑似コードを図 2 に示す．図中，MaxCliqueEnum$(\cdot)$ は，引数として与えたグラフのすべての極大クリークを順次出力する手続きである．

なお，当疑似コードは，解の列挙処理全体の流れに重点を置いて記述されたものであり，実際の計算手続きにおいては様々な工夫が可能である．例えば，手続き Expand における $NewExt$ は，$NewExt \leftarrow core_{j_\tau}(Xv \cup (K \cap NewCand))$ としてよい．$NewCand$ を (拡張前の) K 中の頂点に限定することで，j_τ-核演算の対象となる頂点集合がより小さくなり，結果としてより小さ

```
procedure Main(G, τ):
  [Input] G = (V, Γ): an undirected graph.
          τ: an isolation parameter (0 < τ ≤ 1.0).
  [Output] All maximal j_τ-cored cliques.
  [Global Variables] G, τ
  begin
    Sort V in degree decsending order;
    Expand(∅, V, V);
  end

procedure Expand(X, Ext, Cand):
  begin
    K ← Ext \ X;
    NL ← {v ∈ K | tail(X) ≺ v};
    if K = ∅ then
      Output X; // as a fixpoint solution
      return;
    endif
    if NL = ∅ then
      if X is j(x)-cored then
        if SolWithFalseCand(X, K) == true then
          Output X; // as a solution with false candidates
        endif
      endif
      return;
    endif
    for each v ∈ NL
      NewCand ← Cand ∩ Γ(v);
      NewExt ← core_{j_τ}(Xv ∪ NewCand);
      if Xv ⊆ NewExt then
        Expand(Xv, NewExt, NewCand);
      endif
    endfor
  end

procedure SolWithFalseCand(X, K):
  begin
    Let j_K be a function defined as
          j_K(x) = max{0, τ · deg_G(x) − |X|} for each x ∈ K;
    for each Q presented by MaxCliqueEnum(G[K]) then
      if core_{j_K}(Q) ≠ ∅ then
        return false; // X is not a solution
      endif
    endfor
    return true; // X is a solution with false candidates
  end
```

図 2: 極大孤立性クリーク列挙アルゴリズム擬似コード

な $NewCand$ が得られることが期待できる．これは，Xv を拡張する際の候補頂点が減少することを意味し，探索処理全体の効率化に寄与する．

また，手続き SolWithFalseCand において極大クリーク列挙木を用いているが，先に述べた通り，実際は，生成途中段階のクリーク C において，$core_{j_{K(X)}}(C) \neq \emptyset$ になり次第 `false` を返すことで，X が解でないことをより早期に検出可能となる．

7 おわりに

本稿では，孤立性疑似クリークの高速列挙に向けた第一歩として，極大孤立性クリークの全列挙問題に対する高速アルゴリズムについて議論した．クリークの拡張処理が解へ到達するための必要条件を明らかにし，それに基づく枝刈り機構を組み込むことで，無駄な探索枝の展開処理を強力に抑制する．また，偽候補を伴う解の検出法を与えることで，不動点解以外の解も漏れ無く抽出可能である．

著者等は先行研究 [6, 7] において，グラフの構造変化の観点から孤立性クリーク，および，孤立性疑似クリークの抽出を行っている．特に，[6] での解は本稿における抽出対象をより限定したものに相当し，その実験結果から，極大孤立性クリークも相当数存在することが示唆される．一方，先行研究では，可変 j_τ-核性やその性質を陽に利用した探索処理を行っていないことから，提案アルゴリズムの有効性については今後の実験により確認したい．提案アルゴリズムは現在実装の最終段階にあり，実験結果については口頭発表時に報告したいと考えている．

参考文献

[1] Pattillo, J., Youssef, N. and Butenko, S.: Clique Relaxation Models in Social Network Analysis, Handbook of Optimization in Complex Networks: Communication and Social Networks, Springer Optimization and Its Applications 58, pp. 143 – 162, 2012

[2] Tomita, E., Tanaka, A. and Takahashi, H.: The Worst-Case Time Complexity for Generating All Maximal Cliques and Computational Experiments, Theoretical Computer Science 363(1), pp. 28 – 42, Elsevier, 2006

[3] Batagelj, V. and Zaversnik, M.: An $O(m)$ Algorithm for Cores Decomposition of Networks, CoRR 2003., cs.DS/0310049 OpenURL

[4] Ito, H. and Iwama, K.: Enumeration of Isolated Cliques and Pseudo-Cliques, ACM Transactions on Algorithms, 5(4), Article 40, 2009

[5] ジェイ 泓杰・原口 誠・大久保 好章・富田 悦次：j-核性を持つ極大疑似クリークの全列挙，第 13 回情報科学技術フォーラム - FIT 2014, 第一分冊, pp. 33 - 35, 2014

[6] エラウィンディ サラ・原口 誠・大久保 好章・富田 悦次：クリーク全列挙に基づく構造変化検出アルゴリズム，情報処理学会研究報告，Vol. 2011-MPS-087 No. 32, 2012

[7] Okubo, Y., Haraguchi, M. and Tomita, E.: Structural Change Pattern Mining Based on Constrained Maximal k-Plex Search, Proc. of the 15th Int'l Conf. on Discovery Science - DS'12, Springer-LNAI 7569, pp. 284 – 298, 2012