

j -核性を持つ極大疑似クリークの全列挙An Algorithm for Enumerating Maximal j -Cored Pseudo Cliques

ジェイ 泓杰[†] 原口 誠[†] 大久保 好章[†] 富田 悦次[‡]
 Hongjie ZHAI Makoto HARAGUCHI Yoshiaki OKUBO Etsuji TOMITA

1. はじめに

グラフやネットワークにおいて、クリーク (完全な部分グラフ) は最も密なコミュニティを表現しており、実用上十分高速なクリーク列挙器が提案されている [1, 8]. 一方、現実のコミュニティでは、隣接関係に関して少数の例外を許した疑似クリークも必要となり、クリークの緩和モデルとしてこれまで様々ものが考察されてきた [2]. 本稿では、そうした緩和モデルの一つである k -Plex [3, 5] を発展させた新たな疑似クリークを提案し、その列挙法について考察する.

k -Plex とは非接続数上限制約 (各頂点 v に対し、 v も含めて非接続な他の頂点は高々 k 個以下) を満たす無向グラフの頂点集合であり、特に $k=1$ の場合はクリークとなる. 非接続数上限を与える k の値がごく小さな場合は、クリーク列挙器に準じる実行時間を期待でき、多くの先行研究では $1 \leq k \leq 5$ 程度の値に対してその有効性を論じてきた. 一方、例えば 20 個の頂点からなる 8-Plex では、各頂点は少なくとも 12 個の頂点と隣接しており、十分に密結合な頂点集合と主張できるだろう. こうした頂点集合を求めるためには、 k の値をより大なもの (この場合 $k=8$) に設定する必要があり、従来の k -Plex では現実的な計算時間で求めるのは困難となる. その理由を一言で述べれば、サイズが k 以下の任意の頂点集合は (その集合内の接続関係の有無に依存せずに) 自明に k -Plex となり、接続情報を有効に利用できないからである. この問題は、連結 k -Plex に限定したとしても解消できない. たとえば、長さが k の鎖は自明に k -Plex となるが、多くのネットワークにおいてそうした鎖の数は膨大であり、「長すぎる」鎖を排除する制約が欲しい.

本稿ではこうした立場から、連結 k -Plex に第 3 の制約、すなわち、 j -核性制約を満たすことを要請する.

j -核性制約 頂点集合 X は、その任意の頂点の X の誘導グラフにおける次数が j 以上のとき、 j -核性制約を満たすという. また、この制約を満たす最大の頂点集合を j -核と呼ぶ.

j -核性は、非接続数上限制約である k -Plex 性とは対照的に、接続数下限制約である. ここで、元のグラフの次数がたとえ j 以上の頂点であっても、次数が j 以上の他の頂点との接続数が j 未満であれば j -核からは除外され、しかも j -核の補集合に属する頂点の除去は再帰的に繰り返される.

[†]北海道大学大学院情報科学研究科
 Graduate School of Information Science and Technology,
 Hokkaido University

[‡]電気通信大学先進アルゴリズム研究ステーション
 The Advanced Algorithms Research Laboratory,
 The University of Electro-Communications

k -Plex の生成過程において、既存の k -Plex X に追加しても k -Plex 性が保存される頂点を候補と呼ぶが、 X から「遠すぎる」ものは候補とはなりえない. よって、遠すぎる頂点への接続を無視することができ、これは候補の次数を減少させることを意味する. このことにより、 j -核性 k -Plex に属する可能性がないものを j -核計算により除去し、候補をさらに絞り込むことが可能となる. 本稿では、こうした j -核性に基づく候補の絞り込みに関するいくつかの性質を明らかにし、 j -核性を持つ極大連結 k -Plex の列挙法について論じる.

2. 基本事項

ここでは、単純無向グラフ $G = (V, E)$ を考える. $v \in V$ の隣接頂点集合を $\Gamma(v)$ と記し、簡単のために $x \notin \Gamma(x)$ を仮定する. G の頂点集合 $X \subseteq V$ において、 $G[X] = (X, E \cap (X \times X))$ で定義されるグラフを X による G の誘導部分グラフと呼ぶ. $G[X]$ における $x \in X$ の次数を $\deg_X(x)$ と記す.

3. 連結な k -Plex

定義 (k -Plex) グラフ $G = (V, E)$ の頂点集合 $X \subseteq V$ を考える. ここで、任意の $x \in X$ に対し、 $|X - \Gamma(v)| \leq k$ となる時、 $G[X]$ (または単に X) を k -Plex と呼ぶ. 特に X が連結なとき連結 k -Plex と呼ぶ.

k -Plex X に対し、 Xx も k -Plex となる $x \notin X$ を X の (k -Plex) 候補と呼び、候補集合を $Cand(X)$ と記す. 極大 k -Plex X は部分 k -Plex X_j にその候補 $x_{j+1} \in Cand(X_j)$ を追加する操作を繰り返すことにより構成できる. 極大 k -Plex 枚挙器は、 k -Plex 列 $\{X_j\}_j$ を探索パスとして持つ探索木の展開により、全ての極大 k -Plex を重複なく列挙することで実現できる.

$$\emptyset = X_0 \subset X_1 = X_0x_1 \subset X_2 = X_1x_2 \subset \dots \subset X_n = X_{n-1}x_n = \{x_1, \dots, x_n\} \quad k\text{-Plex 増加列}$$

一方、連結 k -Plex X の場合は連結部分集合列の構成法が基本となる.

$$\emptyset = X_0 \subset X_1 = X_0x_1 \subset X_2 = X_1x_2 \subset \dots \subset X_n = X_{n-1}x_n = \{x_1, \dots, x_n\} \subset \dots \subset X$$

ただし、 $\Gamma(x_n) \cap X_{n-1} \neq \emptyset$.

連結集合増加列において、 X が k -Plex の場合は、列中の連結集合 X_j は全て k -Plex である. すなわち、 x_{j+1} は X_j の k -Plex 候補でもある. よって、連結 k -Plex の探索においては、 k -Plex の候補で X_j と直接隣接した k -Plex 候補だけを選択・追加するだけで良い. 出力解として非連結なものも許す列挙手法と比較すると大幅な効率化を図れるが、節 1 でも述べたように、一

般に膨大な数の解を許してしまう。次節では j -核性制約により可能な解の数を削減し、より効率的な疑似クリークの列挙法について考察する。

4. j -核性 k -Plex

定義 (j -核性極大連結 k -Plex)

グラフ $G = (V, E)$ の頂点集合 $X \subseteq V$ を考える。ここで、任意の頂点 $v \in X$ に対して、 $deg_{G[X]}(v) \geq j$ のとき、 X は j -核性を有するという。また、 j -核性を有する最大の頂点集合を j -核 [4] と呼ぶ。

解： j -核性極大連結 k -Plex X をパラメータ j, k に対する解として定める。

まず、サイズが大きな k -Plex は j -核性を持つことを指摘しておこう。

事実1 $|X| \geq j + k$ な k -Plex X は j -核性である。

すなわち、 k -Plex は、サイズが大な場合には妥当なクリーク緩和モデルであることがわかる。問題はサイズが小な k -Plex で生じ、本稿では j -核性によって、疑似クリークとはみなせない極大 k -Plex を排除することになる。

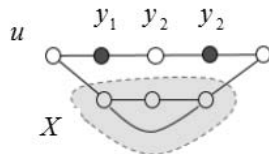
$k \leq |X| < j + k$ の場合、 j -核性までは主張できないが、連結 k -Plex の構成に関して有益な下記の事実が成り立つ。

事実2 $|X| \geq k$ な k -Plex に対し、 $Cand(X) \subseteq D_1(X)$ 、ただし、 $D_\ell(X)$ は X から (最短) 距離 ℓ で到達できる頂点集合を表す。

すなわち、 $k + j > |X| \geq k$ の範囲では、現在の k -Plex X と隣接した頂点のみを、 X に追加可能な候補として考えておけば、連結 k -Plex の構成を目的とする限り十分であることがわかる。さらに、 j -核性まで勘案すると、次の図で示されるように、一部の k -Plex 候補を排除できることがわかる。

$k = 3, j = 2$

u は連結 3-Plex 候補だが、 Xu を含む j -核性 3-Plex は存在しない。

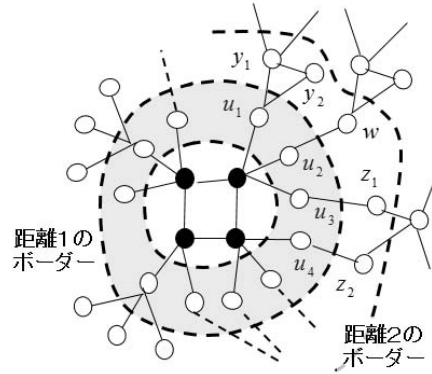


図中、 X から距離 2 以上の点 y_1 等は 3-Plex-候補ではなく、また、3-Plex-候補である u も、距離 1 以下の部分グラフに限定したときの 2-核からは排除され、実質的な候補ではない。つまり、 X は 2-核性の 3-Plex に拡大できない 3-Plex。

事実3 $j + k > |X| \geq k$ なる 連結 k -Plex X に対し、 $Cand(X) \subseteq D_1(X)$ 。さらに、 $y \in Cand(X) - core_j(X \cup Cand(X))$ ならば Xy を含む j -核性 連結 k -Plex は存在しない。ただし、 $core_j(Z)$ は Z の j -核を表す。

上記の事実により、実際の追加可能な候補は k -Plex 候補からさらに絞り込まれ、 j -核性 k -Plex に拡大される可能性を持つものだけに限定できる。

$|X| < k$ の場合は、 k -Plex 候補は距離 1 以内とは限らないが、 $k \leq |X| < k + j$ の場合と同様に、「 X から遠くない点」に限定した誘導グラフにおける j -核を求めることにより、 X を拡大してできる j -核性 k -Plex に含まれることがありえない点を排除することができる。



上図において黒丸●から構成される X に対し、距離が 3 以上の点は、5-Plex 候補にはなりえない。距離が 2 以下の一部の 5-Plex 候補 (w, z_1, z_2 等) は 2-核性から考察外の点だとわかり、これらは除去できる。 w, z_1, z_2 の除去に連動し、 X から距離 1 の一部の点 (u_2, u_3, u_4) も除去できる。これら一連の除去は、 X と X から遠くない点に限定した部分グラフに対して j -核を再帰的に求める形で行われる。図の場合は、 $X \cup \{u_1, y_1, y_2\}$ は j -核の一部となり、よって、 u が実際の追加可能な候補として選択可能となる。下記の事実は以上の議論をまとめたものである。

事実4 連結 k -Plex X で $|X| < k$ なものを考える。

(4-1) X から距離が $k - |X| + 2$ 以上の w に対し、 Xw を含む連結 k -Plex は存在しない。

(4-2) $U(X) = core_j \left(X \cup \bigcup_{i=1}^{k-|X|+1} D_i \right)$ とする。 $w \in Cand(X) - U(X)$ に対し、 Xw を含む j -核性連結 k -Plex は存在しない。 $X \subseteq U(X)$ のとき、 $U_1(X) = U(X) \cap D_1(X)$ が「実際に追加可能な候補」となる。

(4-3) $X - U(X) \neq \phi$ のとき、 X を含む j -核性連結 k -Plex は存在しない。

X と素な $U_1(X)$ の要素 u は、あくまでも Xu を含む j -核性連結 k -Plex が存在する可能性があることのみを保障する。したがって、「実際に追加可能な候補」 u を追加する過程で Xu を含む j -核性連結 k -Plex が存在しないことが判明する場合があります、それは事実 (4-3) の条件、すなわち、 j -核計算による点消去が X の点にまで及ぶ場合である。

以上の議論をまとめよう。 j -核性極大連結 k -Plex Z に至る構成列

$\phi = X_0 \subset X_1 \subset \dots \subset X_{|Z|} = Z$ s.t. $X_i = X_{i-1}x_i$ において、途中の X_i は j -核性とは限らない。実際、 j -核性制約は単調でも逆単調でもなく、本稿においては、「 j -

核性連結 k -Plex に成長する可能性」のある候補のみに絞り込む計算を行っている。また特に、 $U(X_i = X_{i-1}x_i)$ の計算は、 $U(X_{i-1})$ から再帰的に求まることにも注意したい。このために、探索ノードとしての X_i には、 $U(X_i)$ も保持させておく必要がある。

5. 疑似コード

全体の処理の流れを明確にするために、図1で疑似コードを与えておく。このコードにより、全ての j -核性極大連結 k -Plex を重複なく、かつ漏れなく全列挙できる。まず前処理として全体の j -核をとり、その誘導グラフに限定できることに注意する。全体の j -核は複数の連結成分に分解されるので、 j -核性極大連結 k -Plex の生成は j -核の連結成分 C 毎に実行できることに注意する。すなわち、疑似コードへの実質的な入力は連結成分 C である。また、 $U(X), U_1(X) = U(X) \cap D_1(X)$ は、 $|X| < k$ に対し定めたが、 $|X| \geq j+k$ のときは $U(X) = U_1(X) = Cand(X)$ 、また、 $j+k > |X| \geq k$ のときは $U(X) = U_1(X) = D_1(X) \cap core_j(X \cup Cand(X))$ により拡張定義しておく。

また、疑似コードの中ではクリーク探索の高速化に多大に寄与する「右候補枝刈」も導入している。すなわち、重なりあう極大クリーク C_1, C_2 の共通部分 $C_1 \cap C_2$ に表れる頂点は、重なり合わない $C_1 - C_2$ もしくは $C_2 - C_1$ 中の点のみを種にして極大クリークへと成長させることが可能である。 k -Plex に対しても以下で述べるほぼ同様な性質を持つ。

事実5 (右候補枝刈) X を連結 k -Plex, $u \in U_1(X)$, $R_u = \{y \in U(X) \mid y \in \Gamma(u), X - \Gamma(u) \subseteq \Gamma(y)\}$ とする。 R_u の部分 Z で $Z \cup X$ が k -Plex ならば $X \cup Z$ も k -Plex. つまり、極大連結 k -Plex は u を含めて $U_1(X) - R_u$ の候補を追加することにより生成できる。

X において追加すべき候補は $U_1(X) - R_u$ に属する頂点にさらに絞り込まれる。右候補枝刈によって極大 k -Plex の生成が妨げられることはなく、また、求める解は j -核性極大 k -Plex であることから、右候補枝刈は、解の生成に関して安全な枝刈規則である。

6. まとめと今後の課題

本稿では、極大連結 k -Plex の列挙器をベースに、探索の初期段階においても有効な枝刈規則を搭載した j -核性極大連結 k -Plex 列挙のための性質とアルゴリズムについて論じた。パラメータ j, k の設定は、目標とする疑似クリークサイズと枝密度から大まかに定めることができるだろう。何本の枝が欠けて良いか、また逆に、何本の枝が疑似クリークとして必要であるかは、求める疑似クリークのサイズに依存するからである。

単純な極大連結 k -Plex 列挙器と比較すると、サイズが k 未満の頂点集合 X に対し、 j -核性を持つ k -Plex への種となる候補を保持しておく必要があるが、それは連結性を仮定しない場合の k -Plex においても同様であり、 k 値を増加させたときの組み合わせを減少させる効果がより大きいと考えている。さらに、 j -核の計算は $|X| < j+k$ なる X に対して必要となるが、線形時間で処理可能であり、これも候補数を減少させる枝

```

Main(C) { // C は全頂点集合 V の j-核の連結成分
  C において最大次数の点を  $u_C$  とする;
  for ( $v_i \in C$ ) {
    if ( $v_i \in \Gamma(u_C)$ ) continue; // 右候補
    compute  $U(v_i)$ ;
    if ( $v_i \notin U(v_i)$ ) continue; // 事実 (4-3)
    Left = { $v_\ell \mid v_\ell \prec v_i, v_\ell \in U(v_i)$ }; // 左候補:
    // 先行する  $v_\ell$  で  $v_i$  に追加される可能性があるもの
    Expand( $\{v_i\}, U(v_i), Left$ );
  };
};
Expand(X, U, Left) {
   $U_1 = U \cap D_1(X)$ ; //  $|X| \geq k$  のときは自動で  $U_1 = U$ 
  if ( $U_1 = \phi$ ) { //  $U = \phi$  と同値
    if ( $X$  が  $j$ -核性) {output  $X$ ; return;}; else return;
  };
  choose  $u \in U_1$ ; //
   $NLC = U_1 - Left$ ; // 未試行の候補集合
  while ( $NLC - R_u \neq \phi$ ) { // 事実5 の  $R_u$ 
    select  $v \in NLC - R_u$ ;
    compute  $newU = U(Xv)$ ;
    if ( $Xv - newU \neq \phi$ ) { $NLC = NLC - \{v\}$ ; continue;};
    //  $|X| \geq j+k$  のときは  $j$ -核計算と上記の if 文は不要
    Expand( $Xv, newU, Left \cap newU$ );
    Left = Left  $\cup \{v\}$ ;  $NLC = NLC - \{v\}$ 
  };
};

```

図1: j -核性極大連結 k -Plex 全列挙

刈り効果の方が勝ると考えている。実際の検証結果については口頭発表時に報告したい。

参考文献

- [1] 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.
- [2] Pattillo, J., Youssef, N. and Butenko, S.: Clique Relaxation Models in Social Network Analysis, Thai, M. T. and Pardalos, P. M. (eds.), Handbook of Optimization in Complex Networks: Communication and Social Networks, Springer Optimization and Its Applications 57, pp. 143 - 162, Springer, 2012.
- [3] Seidman, S. B. and Foster, B. L.: A Graph Theoretic Generalization of the Clique Concept, Journal of Mathematical Sociology 6, pp. 139 - 154, Taylor and Francis, 1978.
- [4] Seidman, S. B.: Network Structure and Minimum Degree, Social Networks, 5, pp. 269 - 287, 1983.
- [5] Wu, B. and Pei, X.: A Parallel Algorithm for Enumerating All the Maximal k -Plexes, Proc. of the PAKDD 2007 Workshops, LNAI-4819, pp. 476 - 483, 2007.
- [6] Batagelj, V. and Zaversnik, M.: An $O(m)$ algorithm for cores decomposition of networks, Advances in Data Analysis and Classification, Vol 5, pp. 129-145, Springer, 2003.
- [7] Okubo, Y., Haraguchi, M. and Tomita, E.: Relational Change Pattern Mining Based on Modularity Difference, Proc. of the 7th Multi-Disciplinary International Workshop on Artificial Intelligence - MI-WAI'13, LNAI-8271, pp. 187 - 198, (2013)
- [8] Eppstein, D. and Strash, D.: Listing All Maximal Cliques in Large Sparse Real-World Graphs, Proc. of the 10th Int'l Symposium on Experimental Algorithms - SEA'11, LNCS-6630, pp. 364 - 375 (2011)