Vol.2015-MPS-106 No.6
2015/12/15

大規模グラフにおける 擬似クリーク厳密解全列挙に関する考察

大久保 好章[1]　ジェイ 泓杰[1]　原口 誠[1,a)]　富田 悦次[2]

概要：本報告では，著者等が先に提案した，所与のグラフにおける擬似クリークの全列挙問題に対する高速アルゴリズムの有効性を実験的に検証する．抽出対象とする擬似クリークは，それを構成する各頂点毎に非隣接上限数を定めた k-Plex モデルに対し，十分な密度を保証するための隣接下限数を新たな制約として課した j-核性 k-Plex であり，j-核計算の単調性を利用することで，探索処理における不要な探索枝の展開が強力に抑制される．実験では，80 万超頂点のウェブグラフを含むネットワークに対して，最新の極大 k-Plex 列挙システムとの比較を行い，計算時間と解の品質の観点から本システムの有効性を確認する．

1. はじめに

ソーシャルネットワーク解析におけるコミュニティ抽出において，クリークを形成する頂点集合は，理想的なコミュニティ構造であり，そこに観測される逆単調性に基づく効率的な全列挙が実現されている [6]．しかし，現実のコミュニティがクリークとして出現することは極めて稀であり，様々なコミュニティを抽出する観点からは強過ぎるモデルと言える．こうした背景のもと，これまでに様々な擬似クリークが提案された [5]．

一方，別のアプローチとして，グラフクラスタリング (例えば [4]) が挙げられる．そこでは，通常，グラフ全体を俯瞰するために，サイズの大きな少数のクラスタ抽出が想定され，小規模なコミュニティはそれらに吸収されて識別不能となる．よって，比較的サイズが小さなコミュニティを抽出する場合は，(擬似) クリーク列挙器が適している．

密度に基づく擬似クリークモデル (例えば [3]) では，頂点集合が与える部分グラフの密度を考慮するが，そこでは一般に逆単調性が成り立たないことから，その列挙法の多くは経験則に基づくものとなり，完全性を持たない．さらに，例え完全な列挙器があったとしても，解の総数が膨大となることから，大規模グラフを処理することは事実上不可能と考えられる．こうした難題はあるものの，大規模ネットワークに対して実用的に動作する完全な列挙器を開発することは重要であり，それはコミュニティ抽出のみならず，(擬似) クリークに関する統計的な性質をもとにネットワークの特性を解析する際にも有用なツールとなる [8]．

他のクリーク緩和モデルとして，頂点間距離に基づく k-clique k-club，および，k-clan が提案されている [5]．そこでは，許容される頂点間距離をパラメータ k により制御するが，k が大きい場合，元のグラフでは密とは考え難い部分グラフを含む頂点集合が抽出される問題がある．

一方，k-plex モデル [2] は，その頂点集合内での非隣接頂点数の上限を定めることで，元のグラフにおける密度を考慮する．さらに，クリークと同様，k-Plex モデルでは逆単調性が成り立つことから，その列挙器の設計においても好都合である．こうした理由から，著者等は文献 [10] で k-Plex について議論し，以下で述べる欠点を補うための新たな制約を導入することで，抽出すべきターゲットをより洗練化した．

頂点集合 X について，その任意の頂点 x が高々 k の X 中の頂点と非隣接である時，X は k-Plex と呼ばれる．すなわち，非隣接数の上限がパラメータ k で与えられるが，その定義上，非連結な頂点集合が k-Plex となり得る．しかし，こうした非連結な頂点集合は，コミュニティとして不適切であることから，議論からは除外する．

所与の k に対して，サイズが大きな連結 k-Plex は十分な密度を有することが期待できるが，比較的サイズが小さな連結 k-Plex の中には低密度なものも多数含まれる．別の言い方をすると，抽出すべき擬似クリークのサイズと密度を考慮できる何らかの制約が必要となる．つまり，サイズ n の k-Plex を考えると，その各頂点は少なくとも $n-k$

[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
[a)] mh@ist.hokudai.ac.jp

の頂点と隣接するが，この $n-k$ は，ユーザが密な連結頂点集合として許容可能な値であることが望まれる．これに対し，著者等は文献 [10] で，それを構成する各頂点に，隣接頂点数の下限 (*Connection Lower Bound:CLB*) を要請するための新たなパラメータ j を導入し，CLB 制約を満たす連結な極大 k-Plex の全列挙問題を定式化した．さらに，こうした問題に対し，頂点集合の j-核性の概念を用いた考察を通して，解に到達できる可能性のない膨大な探索ノードの展開を抑制可能な高速深さ優先列挙アルゴリズムを提案している．

本稿では，著者等が先に提案した高速アルゴリズムの有効性を実験的に検証する．特に，まだ検証が不十分であった大規模グラフにおけるパフォーマンスを，計算時間と解の品質の観点から観察し，その実用的な有効性を確認する．

以下では，まず，文献 [10] の議論に基づき，CLB 制約を満たす連結極大 k-Plex の列挙問題，および，実験システムの基礎となるその高速アルゴリズムについて紹介する．

2. 準備

$V=\{v_1,...,v_{|V|}\}$ を頂点集合，$\Gamma(v_n)$ を頂点 $v_n \in V$ の隣接頂点集合とする単純無向グラフを (V,Γ) と表す．$i<j$ なる任意の頂点対 v_i と v_j について，$v_i \prec v_j$ なる V 上の順序 $\prec$ を与え，v_i は v_j に先行するという．なお，頂点の識別子が不要な場合は，単に v,x,u 等と表記する．

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．頂点 $x \in X$ について，$\Gamma(x) \cap X$ を $\Gamma_X(x)$ と表し，$|\Gamma_X(x)|$ を $deg_X(x)$ で参照する．

頂点集合 X の任意の頂点 $x \in X$ について，$|X-\Gamma_X(x)| \leq k$ である時，X を k-Plex と呼ぶ．定義より，k-Plex Y の任意の部分集合 $X \subset Y$ もまた k-Plex となる．

k-Plex X に頂点 $y \notin X$ を加えた $X \cup \{y\}$ が k-Plex である時，y を X の k-Plex 候補と呼ぶ．X のすべての k-Plex 候補を $Cand(X)$ と表記する．なお，誤解のない範囲で $X \cup \{y\}$ を Xy と略記する．以下では，連結な k-Plex を議論の対象とし，それを c-k-Plex と略記する．

頂点集合 X と頂点 y について，X と y 間の距離を，G における X から y への最短パス長と定め，$dist(X,y)$ で参照する．特に，$x \in X$ の場合は，$dist(X,x)=0$，また，X と y が非連結の場合は，$dist(X,y)=\infty$ とする．

$D_n(X)$ を $\{y \in V \mid dist(X,y)=n\}$ なる頂点集合とする．k-Plex X について，X に直接隣接する k-Plex 候補の集合，すなわち，$K_1(X)=D_1(X) \cap Cand(X)$ を，X における K_1-候補と呼ぶ．

3. 極大連結 k-Plex

極大 c-k-Plex (k-MPC) とは，c-k-Plex の中で集合の包含関係のもとで極大なものを指す．c-k-Plex X は，$K_1(X) \neq \emptyset$ の時，かつ，その時に限り，それを包含する c-k-Plex へ拡張可能であり，具体的には，X における K_1-候補を X に追加することで拡張処理が行われる．

索引 f による c-k-Plex X の構成とは，頂点 $v_{f(i)}$ の列，$(v_{f(1)},v_{f(2)},...,v_{f(|X|)})$ である．ここで，$f(i)$ は，頂点集合としての $X=\{v_{f(1)},...,v_{f(|X|)}\}$ を得る際に i 番目に追加された頂点の識別子であり，拡張処理の各段階における $X_i^f=\{v_{f(1)},...,v_{f(i)}\}$ は c-k-Plex でなければならない．すなわち，

$$v_{f(i+1)} \in K_1(X_i^f) \tag{1}$$

である．X の構成を，列 $(X_1^f,...,X_{|X|}^f)$ で参照する場合もある．任意の k-MPC Z について，その構成は必ず $K_1(Z)=\phi$ なる $Z=Z_{|Z|}^f$ が終端となる．

k-MPC Z について，それを構成する頂点の追加順序に応じて複数の構成が存在する．実際，任意の初期頂点 $v_{n_1} \in Z$ に対して，先に追加された頂点に隣接する頂点 $v_{n_i} \in Z$ の追加を繰返すことで，途中段階の $X_i(\subseteq Z)$ は必ず c-k-Plex となり，$f(i)=n_i$ で与えられる 索引 f による Z^f は Z の構成となる．多くの不要な構成を除外するために，5 節では，十分な密度が保証される k-MPC の構成のみを探索する制御ルールを導入する．

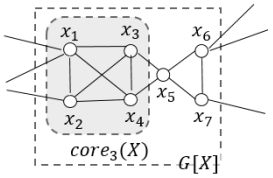

図 1　$j=3$ における j-核計算

4. j-核性 k-MPC

本節では，本稿で抽出ターゲットとする密な k-MPCs を定め，それらの効率的かつ完全な列挙法について述べる [10].

まず，頂点集合の密度を保証する概念として j-核性を導入する．所与のグラフ $G=(V,\Gamma)$ の頂点集合 $X \subseteq V$ について，各頂点 $x \in X$ が $deg_X(x) \geq j$ を満たす時，X は j-核性を有するという．j-核性を有する V の最大部分集合を G の j-核と呼び，$core_j(V)$ で参照する [9]．頂点集合 $X \subseteq V$ について，$core_j(X)$ は $G[X]$ の j-核を意味する．

頂点集合の j-核計算は次の単調性を有する．

事実 (j-核計算の単調性): $X_1 \subseteq X_2$ 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．

X の j-核を求める手続きは単純であり，具体的には，次数が j 未満の頂点を X から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．

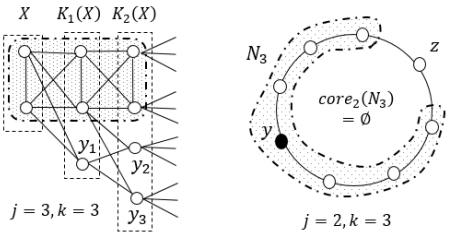

図 2 j-核の有用性

図 1 に j-核計算の例を示す．X の j-核を求める際は，部分グラフ $G[X]$ における次数を考えることから，X の外部との隣接関係は無視される．図中の X において，次数が 3 に満たない x_6 と x_7 がまず除去される．それにより，x_5 の次数が 2 に減少し，新たに除去可能となる．残った v_1, v_2, v_3, v_4 の次数はいずれも 3 となり，除去可能な頂点が存在しないことから，これらが $core_3(X)$ を形成する．

上記をもとに, 本稿における抽出ターゲットを『j-核性を有する k-MPC』と定め，これを j-核性 k-MPC ((j,k)-MPCs) と呼ぶ．以下，所与のグラフにおける j-核性 k-MPC を全列挙する効率的なアルゴリズム [10] について述べる．

k-Plex とは異なり，j-核性を有する頂点集合は一般に逆単調性を持たないが，ここでは，j-核性を有する k-Plex に拡張可能な c-k-Plex の性質をもとに，j-核性 k-MPC の構成探索過程で不要な c-k-Plex を除外する．

事実 (有望な c-k-Plex): ある (j,k)-MPC へ拡張可能な任意の c-k-Plex を X とする．この時,

$$X \subseteq U(X), \quad \text{ここで} \tag{2}$$

$$U(X) = core_j(X \cup Cand(X)). \tag{3}$$

c-k-Plex X が (2) を満たす時，X は有望であるという．逆に，(2) を満たさない X はいかなる (j,k)-MPC へも拡張できない．すなわち，次を満たす X は不要なものとして探索過程において除外可能である．

$$(\textbf{不要性}) \quad X - U(X) \neq \phi \tag{4}$$

j-核演算を適用しない $Cand(X)$ は，サイズ k 未満の X に対する候補の定義としては弱過ぎる．こうした X については，X 中の頂点と候補間のパス長とは無関係に，多数の頂点が候補となる．X をより離れた頂点 y で拡張すると，拡張後の頂点集合内での X 中の頂点と y 間の最短パスはより長くなり，最短パス上の多くの頂点が y と隣接しないことから，拡張後の頂点集合は k-Plex にはならない．X を含む連結 k-Plex を抽出ターゲットとする場合，こうした距離制約は，(j,k)-MPCs を得るための潜在的な候補集合 $U_1(X) = U(X) \cap D_1(X)$ を定める際の鍵となる．

図 2 は，距離制約のもとでの j-核の有用性を示す例である．右図において，黒丸頂点 y のみから成る Y について，$Y \cup Cand(Y)$ は $j = 2$ の場合の j-核性を有する頂点集合全体となる．よって，$Y \cup Cand(Y)$ の j-核計算において削除される頂点は存在しない．しかし，頂点 z について，$dist(Y,z) = 4 > k = 3$ であるから，z と Y の両者を含む c-3-Plex は存在しない．そのため，Y を拡張して j-核性を有する c-3-Plex が得られるかを試みる前に，z を除外する．Y を含む残りの頂点集合は N_3 となるが，$core_j(N_3) = \phi$ であり，このことは Y を含む j-核性を有する c-3-Plex が存在しないことを意味する．

一方，左図は $|X| = 2$ の場合の例である．$dist(X,z) > 3$ なる頂点 z は X を含む c-3-Plex の要素にはなり得ない．これらを除外した後，その j-核を計算すると，最初に y_2, y_3 が削除され，それに伴い y_1 も削除される．点線内の頂点集合は 3-核性を有し，今の場合は極大 3-Plex でもある．

4.1 小サイズの c-k-Plex

$|X| < k$ なる c-k-Plex X を，小サイズの c-k-Plex と呼ぶ．ここで，小サイズの連結な頂点集合 X は c-k-Plex であることに注意すると，$Cand(X)$ は $V - X$ で与えられ，一般に多数の頂点を含む．そこで，不要な候補頂点を除外するために，小サイズの c-k-Plex X と頂点 $y \notin X$ について，$Xy \subseteq Z$ なる c-k-Plex Z が存在するか否かを考える．

$dist(X,y) = 1$ の場合，$Z = Xy$ は明らかに c-k-Plex であるから，ここでは $\ell = dist(X,y) \geq 2$ の場合について考えればよい．部分グラフ $G[Z]$ における X から y への最短パス $p = (y_0, y_1, ..., y_{\ell'} = y)$ を考える．ここで，$y_0 \in X$，および，$y_1, ..., y_{\ell'} \in Z$ である．すると，$\ell \leq \ell'$ であり，y は $X, y_1, ..., y_{\ell'-2}$，および，$y_{\ell'} = y$ それ自身とは隣接しない．Z は k-Plex であるから，$|X| + (\ell' - 2) + 1 \leq k$ でなければならない．よって，$\ell \leq \ell' \leq k - |X| + 1$ であることがわかる．このことは，$dist(X,y) > k - |X| + 1$ なる頂点 y は，X を含む c-k-Plex には決して含まれないことを意味する．これより，小サイズの X について，より精密な $U(X)$ を次の通り与えることができる．

$$U(X) = core_j(X \cup K(X)),$$
$$K(X) = \bigcup_{i=1}^{k-|X|+1} D_i(X), \text{ここで } k - |X| + 1 \text{ は距離制約.}$$

X の拡張に伴い $k - |X| + 1$ は減少することから，$K(X)$ は単調に減少する．

4.2 中間サイズの c-k-Plex

$k \leq |X| < j + k$ なる c-k-Plex X を，中間サイズの c-k-Plex と呼ぶ．中間サイズの c-k-Plex X について，X からの距離が 1 より大きな任意の頂点は，X 中の少なくとも k の頂点と非隣接であるから，k-Plex 候補にはなり得ない．よって，$Cand(X) = K_1(X)$, および，$U(X)$ は次の通り与えられる．

$$U(X) = core_j(X \cup K_1(X)).$$

4.3 大サイズの c-k-Plex

$|X| \geq j+k$ なる c-k-Plex X を，大サイズの c-k-Plex と呼ぶ．大サイズの X については，$j+k \leq |X|$，および，$|X - \Gamma_X(x)| = |X| - |\Gamma_X(x)| \leq k$ より，$j \leq |\Gamma_X(x)|$ であるから，X は必ず j-核性を有する．すなわち，j-核計算は不要であり，$U(X) = X \cap Cand(X)$ となる．これより，U の更新規則は次の通りとなる．

$$U(Xu) = U(X) \cap Cand(Xu), \text{ ここで } u \in U_1(X).$$

4.4 (j,k)-MPCs の構成

X のサイズに依存して定義される $U(X)$ に基づき，(j,k)-MPC Z の構成 $Z^f = (v_{f(1)}, ..., v_{f(|V|)})$ は次を満たす．

$$U(Z^f_{i+1} = Z^f_i v_{f(i+1)}) \subseteq U(Z^f_i) \quad (5)$$

$$v_{f(i+1)} \in U_1(Z^f_i), \text{ ここで } U_1 \text{ は } |Z^f_i| \text{ に依存}, \quad (6)$$

$$U_1(Z = Z^f_{|Z|}) = \phi. \quad (7)$$

$U_1(Z^f_i) \subseteq K_1(Z^f_i)$ より，(6) は (1) より強い．条件 (7) は，$U_1(Z) \subseteq K_1(Z)$，および，$K_1(Z) = \phi$ より得られる．

5. 探索制御規則

c-k-Plex Z はより密に結合した構造であることから，Z の可能な構成の総数は増大する．不要，かつ，重複した構成の生成を回避する効率的な探索を実現するために，ここではふたつの探索制御規則を導入する．

右候補制御 (RCC と略記) により，多くの不要な構成を除外することができる．構成 $Z^f = (v_{f(1)}, \ldots, v_{f(|Z|)})$ は，各 Z^f_i を $U_1(Z^f_i)$ 中の頂点で拡張することで得られる．すなわち，構成の完全列挙を実現するには，$U_1(Z^f_i)$ 中の任意の頂点による Z^f_i の拡張を試みる必要がある．しかし，それらの一部は非極大な k-Plex に到達してしまう．文献 [6] での議論を拡張することで，Z^f_i にそれらを追加しても極大 k-Plex には到達できない不要な頂点集合 $R(Z^f_i)$ を同定することができる．よって，Z^f_i の拡張処理において実際に用いられる頂点集合は $NR(Z^f_i) = U_1(Z^f_i) - R(Z^f_i)$ で与えられ，これを非右候補と呼ぶ．本稿での RCC は，文献 [7] で議論された RCC の拡張と見做すことができ，具体的には，小サイズでない Z^f_i においても適用可能な点で優れている．なお，詳細については文献 [11] を参照されたい．

RCC に加え，各 (j,k)-MPC に対して唯一の構成を構成可能な左候補制御 (LCC と略記) も利用することができる．LCC は，集合列挙における標準的手法のひとつと考えられ，文献 [6] で議論されたものとほぼ同様である．簡単に述べると，各 Z^f_i を拡張する際，$y \prec v_{f(i)}$ なる頂点 y を考慮する必要はなく，これを左候補と呼ぶ．なぜなら，こうした y を用いて得られる構成は，別の左候補 $\ell (\neq y)$ によって他の Z^f_j を拡張することでも得られるとの意味で重複する．よって，左候補の集合を $L(Z^f_i)$ とすると，Z^f_i の拡張処理において実際に用いられる頂点集合は，$NR(Z^f_i) - L(Z^f_i)$ で与えられる．

```
Main(G^input) {  // G^input = (V^input, Γ^input): input graph
  for (each connected component C of core_j(V^input)) {
    Let G = (C, Γ = Γ_C^input);
    ũ = arg max_{u∈C} deg(u);
    NR = C − Γ(ũ); L = φ; X = φ; U = NR;
    Expand(X, NR, L, U);
  };
}
Expand((X, NR, L, U) { // X: c-k-Plex, NR = NR(X),
            // L = L(X), U = U(X) (depending on |X|)
  U_1 = U ∩ D_1(X); //for non-small case, U_1 = U
  if (U_1 = φ) {
    print X; //maximal j-cored c-k-Plex including (j,k)-MPC
    return;
  };
  for (each v ∈ NR − L) {// the order accords to ≺
    Xnew = Xv; Unew = U(Xnew);
    if (Xnew − Unew ≠ φ) continue;// Xnew is hopeless
    NRnew = NR(Xnew); Lnew = (L(X) ∪ {v}) ∩ Unew;
    Expand(Xnew, NRnew, Lnew, Unew)
  };
}
```

図 3 (j,k)-MPC の列挙アルゴリズム

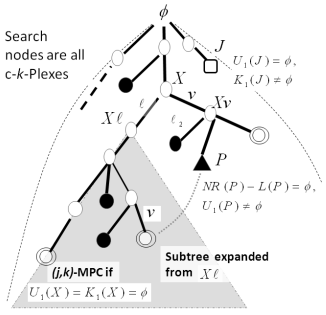

図 4 探索木

6. (j,k)-MPC の列挙アルゴリズム

(j,k)-MPC の列挙アルゴリズムを図 3 に，また，c-k-Plex をノードとする探索木において，手続き $Expand$ が呼び出される様子を図 4 に示す．

空集合に対応する探索木の根ノードから葉ノードに至るパスは，ひとつの c-k-Plex の構成に対応する．黒丸ノードは望みのない c-k-Plex であり，そこからさらに探索枝が張られることはない．一重丸は $NR(X) - L(X) \neq \phi$ 中の頂点で拡張可能なノードであり，二重丸ノードはターゲットとなる (j,k)-MPC である．黒三角ノード P は c-k-Plex であるが，非右候補がすべて左候補となっている．これは，X を拡張して得られる (j,k)-MPC Z は，他の左候補 ℓ によるそのパス上の別の探索枝で生成されることを意味

表 1　各グラフの規模

Name	# of Nodes	# of Edges	Density
WS	50000	500000	0.000400
DBLP	317080	1049866	0.000021
Google	875713	4322051	0.000011

する．別の言い方をすると，Z のある接頭辞 X と非左候補 $\ell \in NR(X)$ について，Z は $X\ell$ を根ノードとする部分木に現れる．白四角ノード J は，k-MPC ではない j-核性 c-k-Plex である．J を含む j-核性 c-k-Plex は存在しないという意味で J は極大である．$K_1(X) = \phi$ なる条件を課すことで，こうした J を除外することは可能であるが，ここでは，すべての (j,k)-MPC に加えて J も出力する．

各 (j,k)-MPC は，グラフの頂点集合 V の j-核，すなわち，$core_j(V)$ の部分集合として現れる．$core_j(V)$ は一般に複数の連結成分から成り，また，ここでのターゲットには連結性を要請することから，$core_j(V)$ の各連結成分 C について，手続き $Expand$ を実行するものとする．

ふたつのパラメータ k と j の設定について，ターゲットの所望のサイズ範囲 $[n_1, n_2]$ と密度パラメータ τ を仮定できる場合は，ターゲット内の隣接次数下限値を $j = n_1\tau$，非隣接次数上限値を $k = n_2 * (1 - \tau)$ とすればよい．

7. 実験

本節では，本アルゴリズムの実装システム JKMPC のパフォーマンスを，計算時間，および，抽出される解の品質の観点から観察する [*1]．

実験では，人工データとして，*Watt-Strogatzs Model* に基づくスモール・ワールドネットワーク WS を生成した．また，実データには，ベンチマークデータである DBLP と Google を用いた [*2]．DBLP は計算機科学分野の学術論文における共著者ネットワークであり，Google は，ウェブページとそれらのハイパーリンクによる接続関係を表したウェブグラフである．これらグラフの規模を表 1 に示す．

著者らが知る限り，(j,k)-MPC の全列挙を主要タスクとするアルゴリズムは存在しない．その意味で，JKMPC と既存システムとの公平な比較は極めて難しい．しかし，本稿のターゲットである (j,k)-MPC は，任意の極大 k-Plex 列挙システムに j-核性判定を組み込むことで抽出可能なことから，ここでは，文献 [1] で提案された最新の極大連結 k-Plex 列挙システム MaxKplexEnum との比較を通して，JKMPC の実用的な有効性を確認する．

所与のグラフについて，MaxKplexEnum はユーザが指定した N 個の極大 k-Plex を列挙する．すなわち，MaxKplexEnum を用いて (j,k)-MPC を全列挙するには，解となるすべての極大 k-Plex を完全に含む適切な N を事前に与える必要がある．言うまでもなく，十分大きな N はそれを可能とするが，同時に，解にはならない膨大な極大 k-Plex も含むことから，現実的ではない．そこで，実験においては，最初に JKMPC を実行することで正確な解総数 $\tilde{N}$ を同定し，MaxKplexEnum が $\tilde{N}$ の極大 k-Plex を抽出する際のパフォーマンスを観察し比較する．こうした設定のもとで MaxKplexEnum が抽出する極大 k-Plex は必ずしもすべての (j,k)-MPC を含むわけではない．その意味で，MaxKplexEnum に最も都合のよい状況を想定した比較であることを強調しておく．

7.1 計算時間

図 5 に，j の値を変動させた場合の両システムによる計算時間を示す．点線は MaxKplexEnum，実線は JKMPC による挙動を表す．また，k の値を変えた結果は，プロットする点の形状 (●, □ 等) の違いで区別している．なお，プロットのない点は，上限 3 時間の計算時間内では解が得られなかったことを意味する．

j の値が大きくなるにつれて，(j,k)-MPCs の総数は減少することから，j が大きな範囲において MaxKplexEnum のタスクはより容易なものとなる．実際，そうした範囲において，MaxKplexEnum は JKMPC よりも高速に動作することが確認できる．しかし，j の値が僅かに小さくなるだけで，そのパフォーマンスは急激に低下し，多くの場合，制限時間内での解の抽出に失敗する．一方，JKMPC のパフォーマンスは概ね安定しており，MaxKplexEnum が抽出に失敗する場合でもすべての解を列挙可能であることがわかる．これより，JKMPC は (j,k)-MPCs の列挙のための効率的かつ実用的なシステムであると考えられる．

7.2 擬似クリークとしての解の品質

k-Plex はクリークの緩和モデルのひとつとして提案されたものであることから，密度の低い極大 k-Plex は望ましいものとは考え難い．こうした観点から解の品質を評価するために，ここでは，MaxKplexEnum と JKMPC により出力される解の密度分布を観察する．

各グラフに対して得られる解の密度分布を図 6 に示す．上段は MaxKplexEnum により得られた解の密度分布，下段は JKMPC によるそれである．WS について，MaxKplexEnum による解は平均 0.80 の密度を示し，標準偏差は 0.14 であった．また，DBLP，および，Google については，それぞれ平均 0.65 と 0.75 の密度を示し，標準偏差は 0.15 と 0.17 であった．これに対して，WS における JKMPC による解の密度の平均は 0.95 であり，標準偏差は 0.018 未満である．さらに，DBLP，および，Google についても，平均の解の密度はいずれも 0.99，標準偏差は 0.0018 未満であった．JKMPC による解が十分な密度を有することは明らかであり，そのすべてを妥当な擬似クリークと考えることに何

[*1] 実装は Java で行い，実行環境は Intel® Core™-i7 (1.7GHz) processor, 8GB memory の PC である．
[*2] http://snap.stanford.edu/data

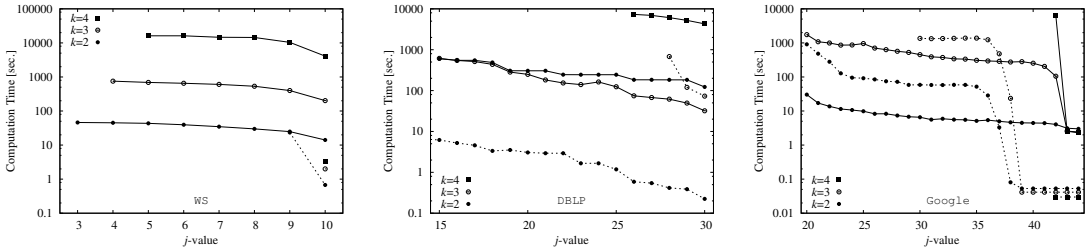

図 5 計算時間

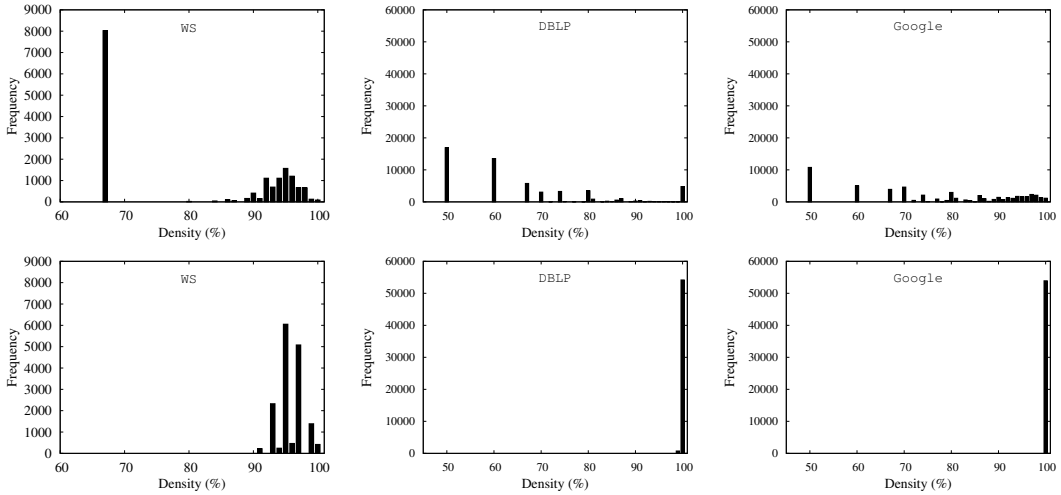

図 6 抽出解の密度分布

ら疑問はないであろう．一方，`MaxKplexEnum` による解には，比較的密度の低い極大 k-Plex が多数含まれており，擬似クリークとしての解の品質が安定していない．このことから，`JKMPC` は，擬似クリーク検出器としても実用的，かつ，有用なシステムであると考えられる．

8. おわりに

本稿では，文献 [10] において提案された (j,k)-MPCs の厳密な全列挙のための深さ優先アルゴリズムについて，その有用性を実験的に検証した．特に，最新の極大 k-Plex 列挙システムとの比較により，一般には困難な擬似クリークの列挙タスクにおいて，頂点数が数十万オーダのグラフにおいても現実的な計算時間でアルゴリズムが動作することを確認した．大規模グラフにおける十分な密度を有する擬似クリークを高速に列挙可能な本アルゴリズムは，実用上極めて有用なツールになるものと考える．

j-核性を考慮することで，列挙すべき解の総数を劇的に減らせるが，グラフが更に大規模化された場合，解総数の問題に再び直面することが予想される．更に解を洗練化するための妥当な制約の考察は今後の重要な課題である．

参考文献

[1] Berlowitz, D., Cohen, S. and Kimelfeld, B.: Efficient Enumeration of Maximal k-Plexes, Proc. of the 2015 ACM SIGMOD Conference, pp. 431 – 444, 2015

[2] Seidman, S. B. and Foster, B. L.:, A Graph-Theoretic Generalization of the Clique Concept, Journal of Mathematical Sociology, 6(1), pp. 139 – 154, 1978

[3] Abello, J., Resende, M. G. C. and Sudarsky, S.: Massive Quasi-Clique Detection, Proc. of LATIN 2002, LNCS-2286, pp. 598 – 612, 2002

[4] Luxburg, U.: A Tutorial on Spectral Clustering, Statistics and Computing, 17 (4), pp. 395 – 416, Springer, 2007

[5] 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

[6] 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

[7] 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

[8] Slater, N. and Itzchack, R. and Louzoun, Y.: Mid Size Cliques are More Common in Real World Networks than Triangles, Network Science, 2(3), pp. 387 – 402, Cambridge Univ. Press, 2014

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

[10] Hongjie Zhai・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

[11] 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, LNAI-7569, pp. 284 – 298, 2012