平成 30 年度
卒業論文

密でないグラフに対して
最大 k-plex を発見する
準指数時間アルゴリズム

名古屋工業大学　情報工学科

所属: 泉研究室

平成 27 年度入学　27115067 佐藤　僚祐

目 次

第1章 はじめに

1.1 研究背景

ソーシャルネットワーク解析等に代表されるように，グラフデータのマイニング分野において，辺密度の高い部分グラフを見つけることは大きな研究のトピックとなっている [1]．辺密度の高い部分グラフとして，最も一般的に使用されているモデルはクリークである. クリークとは全ての 2 頂点間に辺がある部分グラフで, 辺密度最高の部分グラフとみることができる. グラフ中の最も大きいサイズのクリークを見つける最大クリーク問題はグラフアルゴリズムにおける基本的な問題であるが，一般に厳密解，近似解を問わず，その発見問題は NP 完全である．また，クリーク発見における問題として，ほとんど同じ頂点集合からなるクリークを多数発見してしまうことが挙げられる．そこで，クリークを緩和した，高密度部分グラフのさまざまな定義が提案されている. k-plex はそのような定義の一つである．サイズ n の k-plex とは，すべての頂点の次数が $n-k$ であるようなグラフである．$k=1$ のとき 1-plex はクリークと同義である. グラフ中の最も大きいサイズの k-plex を見つける問題を最大 k-plex 問題という．最大 k-plex 問題は最大クリーク問題の一般化であるため自明に NP 完全であり，頂点数 n に対して n の多項式時間で解くことは絶望視されている．そのため，最大 k-plex 問題を高速に解くための指数時間厳密アルゴリズムの研究が行われており，これまでにさまざまなアルゴリズムが提案されている. 提案されたアルゴリズムの多くは $2^{(1-\epsilon)n}n^{O(1)}$ 時間で動作する．ここで ϵ は k の値によって決まる微小定数である．

本研究では，グラフの頂点数でなく，辺の本数をパラメタとした計算時間を持つ厳密アルゴリズムの検討を行う．最大クリーク問題については，辺の本数 m のグラフに対し

て計算時間が $O(2^{O(\sqrt{m})})$ 時間となるようなアルゴリズムが知られている [2] が本研究では最大 k-plex 問題に対して同様のアルゴリズムが設計可能であることを示す．具体的には，辺の本数 m に対して最大 k-plex 問題を $O(n^k 2^{\sqrt{m}})$ 時間で解くアルゴリズムを提案する. このアルゴリズムは m の準指数時間で動くため, 辺の本数が少ないグラフに対する最大 k-plex の発見に有用である.

1.2　関連研究

最大 k-plex 問題を解くためのアルゴリズムとして, 以下のようなものが研究されている.

- 最大 k-plex 問題を整数計画法で定式化して branch-and-cut を用いるアルゴリズムの研究 [3]
- グラフ彩色問題の概念を一般化して k-plex の基数の上限を導いてそれを利用した組み合わせアルゴリズムの研究 [4]
- 最大 k-plex 問題と双対性を持つ d-BDD(Bounded-Degree-d Vertex Detection) 問題の解を見つけるアルゴリズムの研究 [5]

上記の研究で提案されている最大 k-plex 問題を解くためのアルゴリズムは全て $2^n n^{O(1)}$ 時間で動く.

また, 複数のブランチングルールを持った branch-and-search アルゴリズムによって最大 k-plex 問題を $\sigma_k^n n^{O(1)}$ ($\sigma_k < 2$ は k に関する値) で解くアルゴリズムがある. これは 2^n という理論的限界を破った最初のアルゴリズムである. [6]

1.3　本研究の成果

本研究では, 既存の $2^{O(\sqrt{m})}$ 時間の最大クリーク問題のアルゴリズムを応用して最大 2-plex 問題を $n^2 2^{O(\sqrt{m})}$ で解くアルゴリズムを提案する. さらにそのアルゴリズムを拡張させて, 最大 k-plex 問題を $O(n^k 2^{\sqrt{m}})$ で解くアルゴリズムを提案する.

1.4 論文の構成

本論文は全 4 章で構成される. 第 2 章ではグラフの構造と用語の定義をしている. 第 3 章では既存の最大クリーク問題におけるアルゴリズムの導入とそれを応用した最大 2-plex 問題と最大 k-plex 問題のアルゴリズムの提案を行っている. 第 4 章ではまとめと今後の課題について述べている.

第2章

諸定義

2.1　グラフの構造

本論文中のグラフ $G=(V,E)$ は頂点数 $n=|V|$ と辺の本数 $m=|E|$ を持つ単純無向グラフとする. グラフ G の頂点部分集合 S によって誘導される誘導部分グラフを $G[S]$ と表す. 頂点 v に辺が接続されているとき,v に隣接しているといい, v に隣接している頂点を v の近傍と呼び,$N(v)$ と表す. 頂点 (u,v) 間の最短パスの本数を頂点 (u,v) の距離とする.

2.2　最大クリークと最大 k-plex

グラフ G の頂点部分集合 S が完全グラフとき,S をクリークといい, クリークのうち最大サイズのものを見つける問題を最大クリーク問題という. 一方,$G[S]$ の全ての頂点が少なくとも $|S|-k$ の次数を持つとき, S を k-plex といい,k-plex のうち最大サイズのものを見つける問題を最大 k-plex 問題という.

第3章

最大 2-plex 問題と最大 k-plex 問題の準指数時間アルゴリズム

3.1 最大クリーク問題

以下に, 既存の最大クリーク問題に関する定理とその証明を示す. [2]

定理 3.1. m 本の辺を持ったグラフの最大クリーク問題は $2^{O(\sqrt{m})}$ 時間で解くことができる.

証明. G を連結グラフとする. 最小次数の頂点 v を選ぶ. もし v の次数が $\sqrt{m}$ 以上であるならば,

$$2m = \sum_{v \in V} |N(v)| \geqq n\sqrt{m}$$

であるから $n \leqq 2\sqrt{m}$ である. 全ての可能性のある頂点集合を計算するブルートフォースアルゴリズムの実行時間は $n2^{O(n)} = n2^{O(\sqrt{m})}$ となる. 以下 v の次数は $\sqrt{m}$ 未満であるとする.

G 中の最大サイズのクリーク C を見つけるブランチングアルゴリズムを考える. 以下のような 1 つの部分問題にブランチする.

1. v が C 中の頂点の一つである
2. v が C 中の頂点の一つでない

1 つ目の部分問題では $|N(v)|$ の全ての頂点部分集合をブルートフォースによって探索し

て v を含んだ最大サイズのクリークを見つける.2 つ目の部分問題では v を G から削除してアルゴリズムを再帰的に呼び出す.

このアルゴリズムのステップ数が多くとも $n2^{\sqrt{m}}$ になることを m への帰納法によって証明する.

- $m = 0$ のとき,G はただ 1 つのクリークを持ち, その大きさは 1 である.
- $v \in C$ の場合に対応した部分問題を解くために, ブルートフォースを使って $|N(v)|$ 中の最大サイズのクリーク C' を選ぶ.$C = C' \cup v$ である.$|N(v)| < \sqrt{m}$ であるから, 必要なステップ数は多くとも $2^{\sqrt{m}}$ ステップである.
- $v \notin C$ の場合, 帰納法の仮定によって問題は多くとも $(n-1)2^{\sqrt{m}}$ ステップで解ける.

ゆえにアルゴリズムの合計のステップ数は

$$2^{\sqrt{m}} + (n-1)2^{\sqrt{m}} = n2^{\sqrt{m}}$$

でありその実行時間は $2^{O(\sqrt{m})}$ である.

3.2 最大 2-plex 問題

最大 2-plex 問題の準指数時間アルゴリズムを提案するために, 以下に 2 つの補題を示す.

補題 3.1. S を 2-plex の頂点集合とする. ある頂点 v が S に含まれているとき, v から距離が 3 以上離れた頂点は S に含まれない.

証明. v は 2-plex の頂点集合 S に含まれているとする. v から距離が 3 離れた頂点を a,v から距離が 2 離れており a と隣接している頂点を b とする.2-plex の定義より,S 中に含まれる全ての頂点は少なくとも $|S| - 2$ の次数を持つ, つまり S 中に含まれる頂点は S 中の $|S| - 2$ 頂点と隣接しているはずである. もし a が S 中に含まれる頂点であると仮定すると, v は v, a, b と隣接していないため矛盾する. したがって a は S に含まれない. 距離が 4 以上離れている頂点についても同様である.

補題 3.2. S を 2-plex の頂点集合とする. ある頂点 v が S に含まれているとき, v から距離が 2 離れた頂点は 2 つ以上 S に含まれない.

証明. v は 2-plex の頂点集合 S に含まれているとする. v から距離が 2 離れた 2 つの頂点をそれぞれ a,b とする.S 中に含まれる頂点は S 中の $|S|-2$ 頂点と隣接しているはずである. もし a と b が両方とも S 中に含まれると仮定すると,v は v, a, b と隣接していないため矛盾する. したがって a と b は両方とも S に含まれることはない.

これらの補題をふまえ, 最大 2-plex 問題に関する定理とその証明を示す.

定理 3.2. m 本の辺を持ったグラフの最大 2-plex 問題は $2^{O(\sqrt{m})}$ 時間で解くことができる.

証明. G を連結グラフとする. 最小次数の頂点 v を選ぶ. 定理 3.1 と同様の議論により v の次数は $\sqrt{m}$ 未満であるとする.

G 中の最大サイズの 2-plex である S を見つけるために 2 つの部分問題にブランチする.

1. v が S 中の頂点の 1 つである
2. v が S 中の頂点の 1 つでない

1 つ目の部分問題では v から距離 2 以内にあり,S に含まれる可能性のある頂点部分集合を探索して v を含んだ最大 2-plex を見つける. 2 つ目の部分問題では v を G から削除してアルゴリズムを再帰的に呼び出す.

このアルゴリズムのステップ数が多くとも $n^2 2^{\sqrt{m}}$ になることを m への帰納法によって証明する.

- $m=0$ のとき,G はただ 1 つの 2-plex を持ち, その大きさは 1 である.
- $v \in C$ の場合に対応した部分問題を解くために,v を含んだ最大サイズの 2-plex である S を選ぶ. 補題 3.1 より,v から距離 3 以上離れている頂点は S に含まれない. 補題 3.2 より,v から距離 2 離れている頂点は 2 つ以上は S に含まれない. v の近傍の頂点数を M,v から距離 2 離れた頂点数を N とすると,S を選ぶのに必要な頂点数は $2^M \times N$ である.$M < \sqrt{m}$,$N \leqq n$ であるから, 必要なステップ数は多くとも $n2^{\sqrt{m}}$ ステップである.

- $v \notin C$ の場合, 帰納法の仮定によって問題は多くとも $(n-1)n2^{\sqrt{m}}$ ステップで解ける.

ゆえにアルゴリズムの合計のステップ数は

$$n2^{\sqrt{m}} + (n-1)n2^{\sqrt{m}} = n^2 2^{\sqrt{m}}$$

でありその実行時間は $2^{O(\sqrt{m})}$ である.

3.3 最大 k-plex 問題

3.2 では最大 2-plex 問題のついての準指数時間アルゴリズムの存在について述べた. これを応用して一般の最大 k-plex 問題での準指数時間アルゴリズムの存在を示す.

まず, 補題 3.2 を一般の場合に拡張する.

補題 3.3. S を k-plex の頂点集合とする. ある頂点 v が S に含まれているとき, v から距離が 2 以上離れた頂点は k つ以上 S に含まれない.

証明. v は 2-plex の頂点集合 S に含まれているとする.v から距離 2 以上離れた k 頂点 $a_1, a_2, \ldots, a_k$ が全て S に含まれていると仮定する.k-plex の定義より, S 中に含まれる全ての頂点は少なくとも $|S|-k$ の次数を持つ, つまり S 中に含まれる頂点は S 中の $|S|-k$ 頂点と隣接しているはずである. しかし v は $v, a_1, a_2, \ldots, a_k$ の $k+1$ 頂点と隣接していないので, 矛盾する. したがって $a_1, a_2, \ldots, a_k$ は全て S に含まれることはない.

これらの補題をふまえ, 最大 k-plex 問題に関する定理とその証明を示す.

定理 3.3. m 本の辺を持ったグラフの最大 k-plex 問題を $O(n^k 2^{\sqrt{m}})$ 時間で解くアルゴリズムが存在する.

証明. G を連結グラフとする. 最小次数の頂点 v を選ぶ. 定理 3.1 と同様の議論により v の次数は $\sqrt{m}$ 未満であるとする.

G 中の最大サイズの k-plex である S を見つけるために 2 つの部分問題にブランチする.

1. v が S 中の頂点の 1 つである

2. v が S 中の頂点の 1 つでない

1 つ目の部分問題では S に含まれる可能性のある頂点部分集合を探索して v を含んだ最大 k-plex を見つける. 2 つ目の部分問題では v を G から削除してアルゴリズムを再帰的に呼び出す.

このアルゴリズムのステップ数が多くとも $n^k 2^{\sqrt{m}}$ になることを m への帰納法によって証明する.

- $m = 0$ のとき,G はただ 1 つの k-plex を持ち, その大きさは 1 である.

- $v \in C$ の場合に対応した部分問題を解くために,v を含んだ最大サイズの k-plex である S を選ぶ. 補題 3.3 より,v から距離 2 以上離れている頂点は k つ以上は S に含まれない. v の近傍の頂点数を M,v から距離 2 以上離れた頂点数を N とすると,S を選ぶのに必要な頂点数は
$$2^M \times \sum_{i=1}^{k-1} \binom{N}{i}$$
である.$M < \sqrt{m}$,$N \leqq n$ であり, また
$$\sum_{i=1}^{k} \binom{n}{i} \leqq \left(\frac{en}{k}\right)^k$$
であることが知られているので必要なステップ数は多くとも $\left(\frac{en}{k}\right)^k 2^{\sqrt{m}}$ ステップである.

- $v \notin C$ の場合, 帰納法の仮定によって問題は多くとも $(n-1)\left(\frac{en}{k}\right)^k 2^{\sqrt{m}}$ ステップで解ける.

ゆえにアルゴリズムの合計のステップ数は

$$\left(\frac{en}{k}\right)^k 2^{\sqrt{m}} + (n-1)\left(\frac{en}{k}\right)^k 2^{\sqrt{m}} = n\left(\frac{en}{k}\right)^k 2^{\sqrt{m}}$$

でありその実行時間は $O(n\left(\frac{en}{k}\right)^k 2^{\sqrt{m}}) = O(n^k 2^{\sqrt{m}})$ である.

第4章

まとめと今後の課題

4.1 まとめ

今回の研究では, 既存の $2^{O(\sqrt{m})}$ 時間の最大クリーク問題のアルゴリズムを応用して最大 k-plex を解くアルゴリズムを考えた. その結果, 最大 k-plex 問題を $O(n^k 2^{\sqrt{m}})$ 時間で解くアルゴリズムを得ることができた. このアルゴリズムは, 辺の本数に関するパラメータ m の準指数時間で動く. したがって, 辺の本数が少ない, 密でないグラフに対して有効なアルゴリズムである.

4.2 今後の課題

補題 3.2 を応用すると,v から距離 k より離れた点は最大 k-plex の集合に含まれないことが示せる. 3.2 のアルゴリズムでは, 最大 k-plex の集合に含まれることがない頂点も探索しているので, 無駄が存在する. この無駄を解消するために, 頂点を v からの距離によって場合分けして探索範囲を減らすアルゴリズムを考えたが, 計算時間 $O(n^k 2^{\sqrt{m}})$ の境界を破ることはできなかった. v からの距離以外の別のアプローチによって $O(n^k 2^{\sqrt{m}})$ の境界を突破するとこが今後の課題である.

謝辞

本研究の機会を与え，数々の御指導を賜りました泉泰介准教授に深く感謝致します．また，本研究を進めるにあたり多くの助言を頂き，様々な御協力を頂きました泉研究室の学生のみなさんに深く感謝致します．

参考文献

[1] Takashi Washio and Hiroshi Motoda. State of the art of graph-based data mining. *Acm Sigkdd Explorations Newsletter*, 5(1):59–68, 2003.

[2] Fedor V Fomin and Dieter Kratsch. *Exact exponential algorithms*. Springer Science & Business Media, 2010.

[3] Balabhaskar Balasundaram, Sergiy Butenko, and Illya V Hicks. Clique relaxations in social network analysis: The maximum k-plex problem. *Operations Research*, 59(1):133–142, 2011.

[4] Benjamin McClosky and Illya V Hicks. Combinatorial algorithms for the maximum k-plex problem. *Journal of combinatorial optimization*, 23(1):29–49, 2012.

[5] Hannes Moser, Rolf Niedermeier, and Manuel Sorge. Exact combinatorial algorithms and experiments for finding maximum k-plexes. *Journal of combinatorial optimization*, 24(3):347–373, 2012.

[6] Mingyu Xiao, Weibo Lin, Yuanshun Dai, and Yifeng Zeng. A fast algorithm to compute maximum k-plexes in social network analysis. In *AAAI*, pages 919–925, 2017.