平成30年度 情報工学科卒業研究概要		
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	ネットワーク系	泉研究室
	No. 27115067	佐藤 僚祐

1 はじめに

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.

2 諸定義

グラフの構造を次のように定義する. 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.

3 最大 k-plex 問題の準指数時間アルゴリズム

我々が提案する最大 k-plex の発見アルゴリズムは, 最大クリーク問題を $O(2^{O(\sqrt{m})})$ で解くことができるという既存の結果を応用している. 以下に本研究のアルゴリズムとその実行時間を示す.

最小次数の頂点 v を選ぶ ($v<\sqrt{m}$ が仮定できる). 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. 補題により距離 2 以上離れている頂点は k つ以上は S に含まれないことが示せるので, 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:v の近傍の頂点数,N:v から距離 2 以上離れた頂点数).$M<\sqrt{m}$,$N \leqq n$ であり, また $\sum_{i=1}^{k} \binom{n}{i} \leqq (en/k)^k$ 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 $(en/k)^k 2^{\sqrt{m}}$ ステップである.

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. 再帰の回数は多くとも $n-1$ 回である.

ゆえにアルゴリズムのステップ数は

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

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.

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.