Vol. 37 No. 10 情報処理学会論文誌 Oct. 1996

テクニカルノート

定数次数のグラフの最大クリークを抽出するビット演算アルゴリズム

松野浩嗣† 田中都子†,☆

n 個の節点の次数の最大値がある定数でおさえられるような無向グラフ中の最大クリークを，主にビット演算を用いて抽出する時間計算量が $O(n \log n)$ のアルゴリズムを提案し，その正当性について議論する．

A Bit Operation Algorithm for Finding a Maximum Clique in a Constant Degree Bounded Graph

HIROSHI MATSUNO† and MIYAKO TANAKA†,☆

We present an algorithm for finding a maximum clique in an undirectional graph whose maximum size of degrees is bounded by a constant and discuss the validness of the algorithm. The main part of the algorithm consists of bit operations and the time complexity of the algorithm is $O(n \log n)$, where n is the number of vertices in the graph.

1. まえがき

無向グラフ中の最大クリークを抽出する問題はNP完全のクラスに属する問題である[4]．したがってこの問題を厳密に多項式時間内で解くことはほぼ絶望的とみることができるため，指数時間内ではあるができるだけ速く厳密な解を求める[7]，できるだけ良い近似解を多項式時間内に求める[6]といったアプローチがある．このほかに，ある種の制限があるグラフに対して多項式時間のアルゴリズムを与えるというアプローチが考えられるが，これまでに chordal graph について $O(n^2)$ [1]，transitive graph について $O(n^2)$ [2]，circle graph について $O(n^3)$ [3]の時間計算量のアルゴリズムなどが知られている．

本論文はこの3番目のアプローチに基づくものであり，n 個の節点の次数の最大値がある定数でおさえられるような無向グラフについて，その最大クリークを $O(n \log n)$ で抽出するアルゴリズムを提案し，その正当性について議論する．このアルゴリズムはその主要部分がビット演算によって構成されていることが特徴であり，実際の計算機においても高速な計算が期待できる．

† 山口大学理学部
Faculty of Science, Yamaguchi University
☆ 現在，奈良女子大学大学院情報科学専攻
Presently with Graduate School, Nara Wowen's University

2. アルゴリズム

対象とするグラフを最大次数がある定数の単純グラフとし，$G=(V,E)$ で表す．V は n 個の節点の集合であり，$V=\{v_1, v_2, \ldots, v_n\}$ とする．E は枝の集合であり，E の要素を (v_i, v_j) $(=(v_j, v_i))$ と書く．

節点の集合 V のある部分集合 V' に対し，この集合 V' に含まれるどの2つの節点間にも枝があるとき，V' を節点集合とするようなグラフを G の完全部分グラフといい，極大な完全部分グラフをクリークと呼ぶ．便宜上，孤立点は節点数1のクリークとして扱う．グラフ G に含まれるクリークのうち，最大の節点数からなるものを最大クリークと呼ぶ．

V の部分集合 V' $(|V'|=m)$ について，次のような2進配列 B_i と B_j を構成する（任意の集合 Q に対し，$|Q|$ で集合 Q に含まれる要素の個数を表す）．各 v_{ik} は節点 $v_i \in V'$ に対応し，これをラベルとする配列の内容を $[v_{ik}]$ $(=0,1)$ と書く．

$$B_i = [v_{i1}][v_{i2}]\cdots[v_{im}],$$
$$B_j = [v_{j1}][v_{j2}]\cdots[v_{jm}].$$

さらに変数 F を次のように定める．

$$F = [f_1][f_2]\cdots[f_m] \quad (\text{各 } [f_i]=0,1).$$

記号 $\wedge$ で論理積を表す．B_i と B_j について $1 \leq$

$\ell \leq m$ なるすべての ℓ に対し $[f_\ell] \leftarrow [v_{i\ell}] \wedge [v_{j\ell}]$ であるとき，$F \leftarrow B_i \wedge B_j$ と書く．記号 $\vee$，$\oplus$ をそれぞれ論理和，排他的論理和としたとき，$F \leftarrow B_i \vee B_j$，$F \leftarrow B_i \oplus B_j$ も同様に定義される．

以下の CLIQUEWITH(P) は，最大次数が定数 d $(1 \leq d \leq n)$ のグラフ $G = (V, E)$ の任意の 1 節点 $P \in V$ を与え，節点集合 V_P $(|V_P| = p{:}\, p$ は $1 \leq p \leq d$ なる定数) と，P を含むクリークを表す長さ p の 2 進配列の集合 R の組 (V_P, R) を返す手続きである．集合 R の各要素に対し，左から i 番目のビットを i ビット目と呼ぶ（MSB は 1 ビット目，LSB を p ビット目）．ある e $(1 \leq e \leq p)$ および各 j $(1 \leq j \leq e)$ に対し i_j $(1 \leq i_j \leq p)$ 番目のビットが‘1’でその他のビットが‘0’であるとき，アルゴリズム中の集合 V_P に含まれる節点 v_{i_j} $(1 \leq j \leq e)$ および節点 P からなるグラフ G の部分グラフはクリークとなっている．

ただし，前処理として枝集合 E は適当な線形順序によってソーティングされているものとする．グラフ G の最大次数は定数であるので集合 E の要素の個数 $|E|$ は $O(n)$ であり，マージソート，ヒープソートなどの手法によりこのソーティングは $O(n \log n)$ の計算時間で行える[5]．

```
procedure CLIQUEWITH(P):
begin
    V_P={v | (P,v) ∈ E}
       ={v_1, v_2, ..., v_p}  (1 ≤ p ≤ d);
    E_P={(v_i, v_j) | ∀v_i, v_j
       ∈ V_P(1 ≤ i < j ≤ p)[(v_i, v_j) ∈ E]};
    ℓ ←1;
    for i ←1 until p do
        begin
            define the array B_ℓ = [v_ℓ1][v_ℓ2]···[v_ℓp]
            where [v_ℓi] ← 1 and
            [v_ℓh] ← 0  (∀h (1 ≤ h ≤ p) & h ≠ i);
            ℓ ← ℓ + 1;
        end
    q ← 0;
    R ← φ;
    repeat
        begin
            Q ← φ;
            for i ← 1 until ℓ − 2 do
                for j ← i + 1 until ℓ − 1 do
                    begin
                        F ← B_i ∧ B_j ;
                        if (the number of '1' in F is q)
                            then
                            begin
                                F ← B_i ⊕ B_j,
                                where [f_r] = 1, [f_s] = 1
                                for some r, s
                                (1 ≤ r < s ≤ p);
                                if (v_r, v_s) ∈ E_P then
                                    begin
                                        Q←Q ∪ {B_i∨B_j};
                                        R←Q;
                                    end
                            end
                    end
            if |Q| ≥ 2 then
                begin
                    ℓ ← |Q| + 1;
                    for i ← 1 until ℓ − 1 do
                        let the i-th element in Q be B_i;
                        q ← q + 1;
                end
        end
    until |Q| ≤ 1
    return (V_P, R)
end
```

枝集合 E はソーティングされているので，この手続きの先頭部分の 2 つの集合 V_P および E_P は 2 分探索法によってそれぞれ $O(\log n)$ の計算時間で求めることができる（$|E| = O(n)$ であることに注意）[5]．これ以降の部分は，すべて要素数が定数の集合を用いているので，定数の時間で計算できる．よって，手続き CLIQUEWITH(P) の計算時間は $O(\log n)$ である．

以下に定数次数のグラフの最大クリークを抽出するアルゴリズムを示す．アルゴリズム中の COUNTONE((V_c, R_c)) は，集合 R_c に含まれる 2 進配列中に現れる 1 の個数を求める関数である（R_c 中の 2 進配列のすべてが同じ個数の 1 を持つことに注意）．

［アルゴリズム 1］
入力：定数次数のグラフ $G = (V, E)$
出力：グラフ G の最大クリークを表す，節点集合 V_M $(\subseteq V)$ と 2 進配列の集合 R_M の 2 つ組 (V_M, R_M)

```
begin
    枝集合 E をソートする;
    c1 ← 0;
    maxsize ← 0;
    (Vc, Rc) ← (φ, φ);
    (VM, RM) ← (φ, φ);
    for 節点集合 V に属するすべての節点 v do
        begin
            (Vc, Rc) ← CLIQUEWITH(v);
            c1 ← COUNTONE((Vc, Rc));
            if maxsize < c1 then
                begin
                    maxsize ← c1;
                    (VM, RM) ← (Vc, Rc);
                end
        end
end
```

前述のように，枝集合 E のソーティングは $O(n\log n)$ で行える．また **for** ループにおいて，節点集合 V ($|V|=n$) の各々の要素 v に対して，$O(\log n)$ の手続き CLIQUEWITH(v) を実行する．したがって，このアルゴリズムの計算時間は $O(n\log n)$ である．

3. 正当性の証明

手続き CLIQUEWITH(P) 中の 2 進配列がグラフ $G=(V,E)$ の節点 $P\in V$ を含むクリークを表していることを証明する．これにより，アルゴリズム 1 の正当性は容易に確かめられる．まず，次の補題を証明しよう．

補題 1 各 $s\geq 2$ に対し，s 個の節点を持つグラフ G の中に $s-1$ 個の節点からなるクリークが 2 つあるための必要十分条件は，グラフ G が次数 $s-1$ の節点を $s-2$ 個，次数 $s-2$ の節点を 2 個持つことである．

(証明) 孤立点を節点数 1 のクリークとして扱うことに注意すれば，$s=2$ の場合に本補題が成り立つことは明らかであるので，以下は $s\geq 3$ の場合について考える．与えられたグラフを $G=(V,E)$ とし，次数 $s-2$ を持つ 2 つの節点を a，b とする．節点 a に対し $V_a=\{v\mid v\neq b\ \&\ (a,v)\in E\}$ とすると，V_a の各要素である節点は各々次数 $s-1$ を持つ．G の節点数は s であるから，これらの節点は G の他のすべての節点との間の枝を持つ．したがって，$V_a\cup\{a\}$ を節点集合とするグラフはクリークとなっている．同様な議論が節点 b に対してもいえる．以上より，十分条件が成り立つ．

次に必要条件が成り立つことを証明しよう．グラフ G の中の $s-1$ 個の節点からなるクリークの 1 つを $K_{s-1}=(V_{s-1},E_{s-1})$ とする．$V_{s-1}=\{v_1,v_2,\ldots,v_{s-1}\}$ とすると，明らかに V_{s-1} の各節点の次数は $s-2$ である．

G の中には，K_{s-1} に含まれない 1 つの節点 v が存在する．G の中の K_{s-1} でない $s-1$ 個の節点からなるクリーク K'_{s-1} は明らかに節点 v を含むので，ある節点 $w\in V_{s-1}$ について $(v,w)\notin E$ である．

$K'_{s-1}=(V'_{s-1},E'_{s-1})$ とすると，V_{s-1} 中の $(v,v_1)\notin E$ なる節点 v_1 $(=w)$ と節点 v を入れ替えることにより $V'_{s-1}=\{v,v_2,\ldots,v_{s-1}\}$ と書ける．

各 i $(2\leq i\leq s-1)$ に対し，$v_i\in V_{s-1}$ であるので，各節点 v_i，v_j $(2\leq i,\ j\leq s-1,\ i\neq j)$ 間には枝が存在する．したがってこれらの枝の集合を E' とすると $E'_{s-1}-E'=\{(v,v_j)\mid 2\leq i\leq s-1\}$ となる．よって，節点 v の次数は $s-2$ である．

以上より，グラフ G は次数が $s-2$ の 2 つの節点 v,v_1 を含む．さらに，グラフ $(V'_{s-1}-\{v\},E')$ $(E'\subset E'_{s-1})$ が $s-2$ 個の節点からなるクリークになっていること，および $(v,v_1)\notin E$ であることに注意すれば，他の $s-2$ 個の節点の次数がすべて $s-1$ であることも容易に確かめられる．□

定理 1 手続き CLIQUEWITH(v) 中の 2 進配列 B_ℓ (p ビットとする) について，各 $e\geq 1$ に対し B_ℓ に e 個の '1' が含まれているとき，これに対応するグラフは節点数 $e+1$ からなるクリークとなっている．

(証明) e に関する数学的帰納法によって証明する．$e=1$ のときは，最初の **for** ループ内での B_ℓ の構成のしかたよりこれに対応するグラフが節点数 2 のクリークとなっていることは明らかである．

次に，$e=r$ 個の '1' を含む 2 つの p ビットの 2 進配列 B_ℓ，$B_{\ell'}$ が存在し，これに対応するクリークがそれぞれ K_{r+1}，K'_{r+1} であるとしよう．配列 B_ℓ と $B_{\ell'}$ との論理積をとった結果，$r-1$ 個の '1' が残ったとする．この場合に配列 B_ℓ と $B_{\ell'}$ の論理和をとると，'1' の個数は $r+1$ となり，これに対応するグラフの節点数は $r+2$ である．つまり，節点数 $r+2$ のグラフ G の中に節点数が $r+1$ の 2 つのクリーク K_{r+1}，K'_{r+1} が含まれていることになる．

補題 1 よりこのグラフ G は，r 個の次数 $r+1$ の

	1	2	3	4	5	6	7
B_ℓ:	0	1	0	1	0	0	1
$B_{\ell'}$:	0	0	0	1	0	1	1

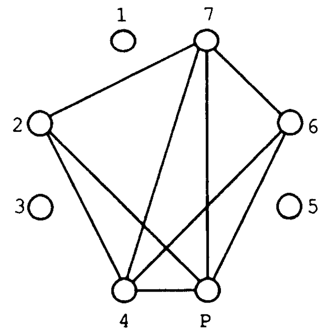

図1 B_ℓ と $B_{\ell'}$ と対応するグラフの例
Fig. 1 Example of B_ℓ and $B_{\ell'}$ and their corresponding graph.

節点と2個の次数 r の節点 a, b からなっていることがいえる（節点 a, b は B_ℓ と $B_{\ell'}$ の排他的論理和をとることで求めることができる）．節点 a, b の間に枝があるかをチェックし，もしあれば B_ℓ と $B_{\ell'}$ の論理和をとったものを新たな p ビットの2進配列としたとき，この配列に対応するグラフは節点数 $r+2$ からなるクリークとなっていることは明らかである． □

図1に配列 B_ℓ と $B_{\ell'}$ とこれらに対応するグラフの例を示す．

4. むすび

本論文では，次数が定数に制限されたグラフの最大クリークを求める $O(n \log n)$ のアルゴリズムを提案し，その正当性について議論した．このアルゴリズムはその主要部分にビット演算を用いているため，実際の計算機においても高速に計算が可能で，ハードウェア化もしやすいと考えられる．今後，このアルゴリズムの応用も含め，実用的な面についての検討を進めて行きたいと考えている．

謝辞 最大クリーク抽出問題に関して多くの貴重なご助言をいただいた，電気通信大学の富田悦次教授に感謝する．

参考文献

1) Gavril, F.: Algorithms for Minimum Coloring, Maximum Clique, Minimum Covering by Cliques, and Maximum Independent Set of a Chordal Graph, *SIAM J. Comput.*, Vol.1, No.2, pp.180–187 (1972).
2) Even, S. and Pnueli, A.: Permutation Graphs and Transitive Graphs, *J. ACM*, Vol.19, No.3, pp.400–410 (1972).
3) Gavril, F.: Algorithms for Maximum Clique and a Maximum Independent Set of a Circle Graph, *Networks*, 3, pp.261–273 (1973).
4) Garay, M.R. and Johnson, D.S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, San Francisco (1979).
5) Aho, A.V., Hopcroft, J.E. and Ullman, J.D.: *Data Structures and Algorithms*, Addison-Wesley, Reading Mass. (1983).
6) 山田義朗，富田悦次，高橋治久：近似最大クリークを抽出する確率アルゴリズムとその実験的評価，電子情報通信学会論文誌，Vol.J76-D-I, No.2, pp.46–53 (1993).
7) 富田悦次，今松憲一，木幡康弘，若月光夫：最大クリークを抽出する単純で効率的な分枝限定アルゴリズムと実験的評価，電子情報通信学会論文誌，Vol.J79-D-I, No.1, pp.1–8 (1996).

（平成8年4月1日受付）
（平成8年7月4日採録）

松野 浩嗣（正会員）

昭和35年生．昭和59年山口大学大学院工学研究科修士課程修了．同年学校法人第二麻生学園山口短期大学講師．昭和62～平成6年大島商船高等専門学校勤務．平成7年4月山口大学理学部自然情報科学科助教授．計算機科学の教育と研究に従事．計算機ネットワークの運用と構築技術にも興味を持つ．電子情報通信学会会員．理学博士．

田中 都子（学生会員）

昭和48年生．平成6年大島商船高等専門学校情報工学科卒業．平成6年山口大学理学部物理学科編入学．平成8年奈良女子大学大学院情報科学専攻入学．山口大学在学中，グラフ理論の研究に従事．