最大クリークを抽出する単純で効率的な分枝限定手法

6F－3

今松 憲一　富田 悦次　小川 剛[†]　木幡 康弘[††]

電気通信大学 電子情報学科　([†]現 TOA, [††]現 NTT データ)

1 まえがき

最大クリークを厳密に見つける方法が幾つか研究されてきているが，部分問題に分割を繰り返し探索する時，最終的な分割数すなわち分枝数と一つの部分問題にかかる手数を含めた結果が実行時間に結び付く．一般に分枝数を出来るだけ限定することが望ましいが，これにかかる費用が大きすぎるとかえって総実行時間が増大する．

本稿では，比較的軽い単純な処理で大きな分枝限定効果を得る手法を提唱し，その実験的評価を行った．

2 アルゴリズム

諸定義および記法は文献 [1] に準ずる．厳密に最大クリークを一つ抽出するためには，逐次クリークを求めていき最終的に最大であったものを出力すればよいが，その過程において，既に求められているものより大きいクリークが得られないときの探索は省略できる．従って，探索に先立って得られるクリークの大きさを見積もることを考える．

節点集合 V，枝集合 E の順序対とするグラフ $G=(V,E)$ において，各節点 $v\in V$ に次のような正整数 $No(v)$ を定義すると，G から得られるクリークの大きさ $\omega(G)\leq\max\{No(v)\mid v\in V\}$ が成り立つ [1]．ここで，節点 v に隣接している節点の集合を $\Gamma(v)$ で表す．また V は順序付き集合とする．

1. $w\in\Gamma(v)$ ならば $No(v)\neq No(w)$
2. $No(v)=k>1$ ならば 各 $j=1,2,\cdots,k-1$ に対し $No(w)=j$ なる $w\in\Gamma(v)$ が存在する

条件 1 は即ち彩色であり，$No(v)$ が色の識別子となる．グラフ彩色問題は NP- 困難であるが特に最適な彩色を行うことが主眼ではなく，高速な近似彩色を用いた NUMBERING-ARRANGED (以下 N-A と略) および最大クリークを抽出する MCLIQ として図 1 にまとめた．なお，N-A での和集合も順序つきとすることにより V の最後尾の要素が最大の No に対応する．

```
procedure MCLIQ(G = (V, E))
  Q := φ      } 大域変数
  Qmax := φ   }
  sort V in non-increasing degrees
  NUMBERING-ARRANGED(V,No)
  EXPAND(V,No)
  output Qmax
end {of MCLIQ}
procedure EXPAND(V,No)
  while V ≠ φ do
    v := vertex in V such that No(v) is greatest
    if |Q| + No(v) > |Qmax| then
      Q := Q ∪ {v}
      V' := Γ(v) ∩ V    ··· 部分問題へ分割
      if V' ≠ φ then
        NUMBERING-ARRANGED(V',No')
        EXPAND(V',No')
      else if |Q| > |Qmax| then Qmax := Q fi
      fi
      Q := Q − {v}
    else goto Exit
    fi
    V := V − {v}
  od
  Exit:
end {of EXPAND}
procedure NUMBERING-ARRANGED(V, No)
  maxno := 1
  C1 := φ
  C2 := φ
  while V ≠ φ do
    v := the first vertex in V
    k := 1
    while Ck ∩ Γ(v) ≠ φ do
      k := k + 1
    od
    if k > maxno then
      maxno := k
      Cmaxno+1 := φ
    fi
    No(v) := k
    Ck := Ck ∪ {v}
    V := V − {v}
  od
  V := C1 ∪ C2 ∪ ··· ∪ Cmaxno
end {of NUMBERING-ARRANGED}
```

図 1: MCLIQ

A Simple and Efficient Branch and Bound Technique for Finding a Maximum Clique

Ken'ichi Imamatsu　Etsuji Tomita

Tsuyoshi Ogawa　Yasuhiro Kohata

University of Electro-Communications

1-5-1 Chofugaoka, Chofu, Tokyo 182, Japan

3　実験的評価

MCLIQ の他にも比較対象として

CP 次数に着目し単純にまとめあげられた R.Carraghan, P.M.Pardalos らのアルゴリズム [2]

BB MCLIQ 同様に近似彩色を用い高密度グラフで特に優位性を示す L.Babel のアルゴリズム [3]

を出来るだけ共通に実働化し, 計算機 SUN SPARC station 10 上で実験的に平均実行時間 (Running time) を測定し節点数 (Vertices)500 迄, 枝存在率 (p)0.2,0.5,0.7 について図 2に示した. なお, 以上 3 アルゴリズムは何れも部分問題への分割方法が同じであるため, この回数すなわち分枝数 (Branches) も測定し図 3にて分枝限定効果の強さを比較した.

測定範囲内では全て MCLIQ が最も実行時間の点で優れている. また実行時間と分枝数の双方の結果により,CP に対する優位性は明確である. MCLIQ と BB に関して, 部分問題一つに費やされる時間の差はすなわち近似彩色に費やされる時間にほかならない. この一例を 表 1に示す. BB で用いられた手法は DS と示してある. 分枝数と彩色精度には相関があるが, 実行時間を短くするためには彩色にも高速性が要求される. この点についても MCLIQ は最も効率の良い結果を得ているといえよう.

4　むすび

厳密に最大クリークを抽出する単純で高速なアルゴリズム MCLIQ を提案し, 類似のアルゴリズムとの実験的な比較を行って優位性を示した.

謝辞　実験の対象としたランダムグラフの整備に尽力して下さった本学情報システム学研究科の奥田達哉氏, 御協力・討論頂いた若月光夫助手, 高橋治久助教授に感謝致します.

参考文献

[1] 富田, 藤井.: 最大クリーク抽出の効率化手法とその実験的評価, 電子通信学会論文誌,J68-D,3,pp.221-228(1985)

[2] R.Carraghan, P.M.Pardalos.: *An exact algorithm for the maximum clique problem*, Operations Research Letters 9,pp.375-382(1990)

[3] L.Babel.: *Finding maximum cliques in arbitrary and in special graphs*, Computing 46,pp.321-341(1991)

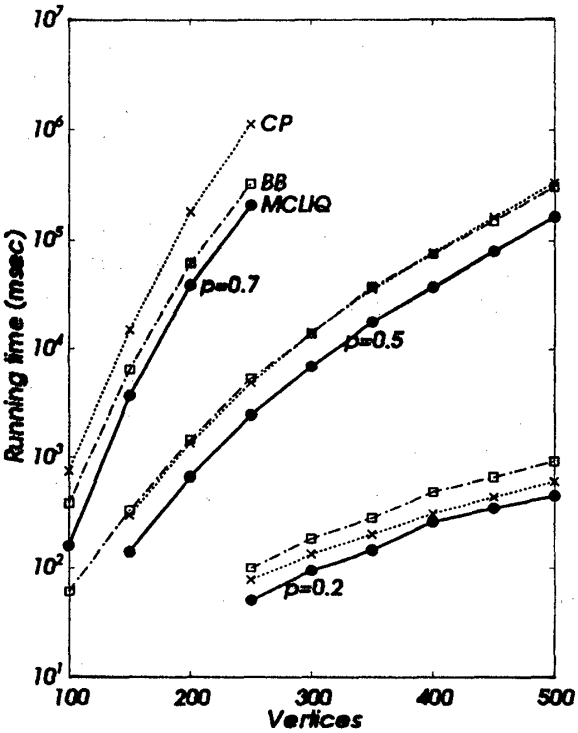

図 2: 平均実行時間 (msec), ランダムグラフ 3 種

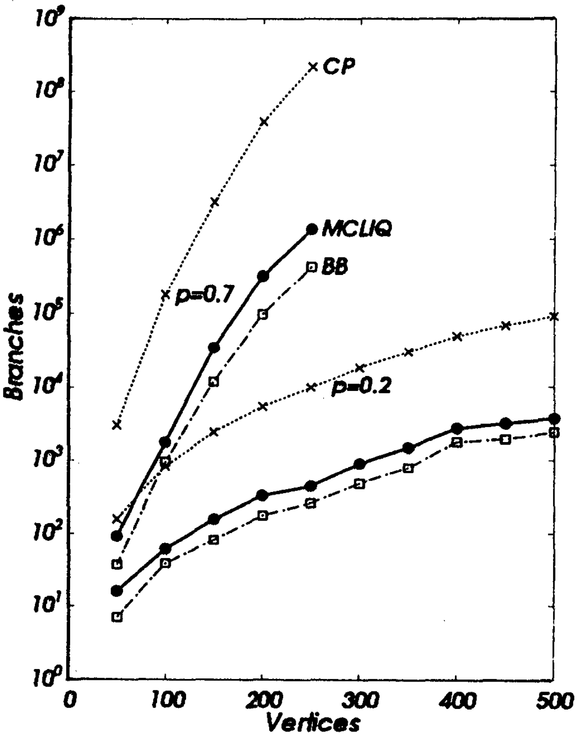

図 3: 平均分枝数, ランダムグラフ 3 種

表 1: 近似彩色,3000 節点ランダムグラフ 5 種

p	N-A		DS	
	colors	(msec)	colors	(msec)
0.2	126-128	289.8	115-117	7309.6
0.5	313-316	473.0	294-298	8256.6
0.7	489-497	679.6	464-472	8866.4