数理モデル化と問題解決　27－9
（１９９９．１１．２６）

最大重みクリークの効率的抽出アルゴリズムとその評価

富田悦次, 若井康, 今松憲一*
電気通信大学 電気通信学部 情報通信工学科
(* 現在, 富士通 (株))

あらまし ある種の数理問題は重み付きグラフの最大重みクリーク抽出問題としてモデル化される. そこで本稿では, 最大重みクリークを効率良く抽出する改良した分枝限定アルゴリズムを提唱する. この分枝限定のためには, 単純な逐次近似彩色を用いている. 本アルゴリズムは, いくつかの重み付きランダムグラフに対する比較実験により評価を行い, 良好な結果を得ている.

An Efficient Algorithm for Finding a Maximum Weight Clique and its Experimental Evaluations

Etsuji Tomita, Yasushi Wakai, and Ken'ichi Imamatsu
Department of Information and Communication Engineering,
The University of Electro-Communications

Abstract Certain problems can be formulated as a maximum weight clique problem in a weighted graph. Then we propose an improved efficient branch and bound algorithm for finding a maximum weight clique. We have shown its effectiveness by computer simulations for several weighted random graphs.

1. はじめに

多くの数理問題がグラフの最大クリーク抽出問題としてモデル化され, いくつかの最大クリーク抽出アルゴリズムが提案されている ([1]〜[3], あるいはその引用文献). 更に, 同問題を一般化したものとして, 各節点に重みが付与されたグラフを対象とし, その中で節点重みの総和が最大になるクリーク (最大重みクリーク) の抽出問題も重要となってきている ([1]). たとえば,RNA の二次構造予測問題の基本部分を最大重みクリーク抽出問題として表現することも出来 ([4]〜[7]), 最大重みクリークを抽出するためのアルゴリズムもいくつか発表されている ([1],[8],[9] 等).

本稿では, 文献 [2] の分枝限定アルゴリズムを基にしたアルゴリズム ([9]) を更に改良し, 効率的な最大重みクリーク抽出アルゴリズムを提唱し, 実験的評価により, その有効性を示す.

ここで, 対象とする無向グラフ G は節点集合 V, 枝集合 E の順序対として $G=(V,E)$ と表す. V 中の各節点 v には実数値の重み $w(v)$ が与えられているとし, $V' \subseteq V$ 中の節点重みの総和を, その集合 V' の重み $w(V') = \sum_{v \in V'} w(v)$ とする. G 中のクリーク $G(Q)$ のうち, $w(Q)$ が最大であるものを最大重みクリークといい, その重みを $\omega(G,w)$ と表す. その他の諸定義及び記法は, 文献 [2],[9] に従う.

2. 分枝限定法

まず, 小さいクリークから出発してより大きいクリークを逐次求めていく手順を, 深さ優先探索により, 最大であると確認できるものが見出されるまで続けていく方法 [2] を基本とする. ここで, 解の探索領域を小さくするために, 何らかの方法を用いて分枝を限定することが効率化の重要な指標となる.

着目節点の隣接節点集合から次々と部分問題である節点集合 V' を構成していくとき, その重み $w(V')$ を求めることにより, 基本的な分枝限定アルゴリズムが得られる.

3. クリーク重みの上界

重みを考慮しない最大クリーク抽出問題に於いて, 文献 [2] で提案されたアルゴリズム MCLIQ は単純な逐次近似彩色を用いることで, 分枝限定を強化して探索分枝数を減少し, かつ, これに伴う計算量の増大を抑えて総実行時間を短縮することに成功している. これを参考にして, 出来るだけ単純かつ効果の大きい分枝限定を考察する.

さて，あるグラフ $G=(V,E)$ を逐次近似彩色（sequential coloring）するには，並べられた節点の先頭から順に，隣接した節点には異なった色番号（正整数）で彩色するとの条件下で，逐次最小の色番号を与えていく．そこで，最大色番号（近似彩色数）が $\tilde{\chi}$ で，色番号 i の節点集合（独立節点集合）が C_i $(i=1,2,\cdots,\tilde{\chi})$ となったとする．ここで，節点 $v\in C_i$ は色番号 $k(v)=i$ をもつという．ここで，

$$z(v)=w(v)+\sum_{i=1}^{\tilde{\chi}}\operatorname*{Max}_{u\in\Gamma(v)\cap C_i} w(u)$$

とおいたとき，$z(v)$ は節点 v を含むクリークの重みの上界を与える．従って，既に見出されている最大のクリーク重み $w(Q_{max})$ および探索過程の完全部分グラフ $G(Q), Q\subseteq\Gamma(v)$ の重み $w(Q)$ について，$z(v)+w(Q)\leq w(Q_{max})$ であるとき，節点 v は $w(Q_{max})$ より重いクリークに含まれ得ない．この時，即座に $V:=V-\{v\}$ として，節点 v を削除してよい．この上界 $z(v)$ を出来るだけ小さく抑えるためには，節点 v に隣接する節点のうち $z(v)$ に与える影響が大きい重みの大きな節点は，できるだけ少ない個数の C_i に集中していることが望ましい．ここで，節点の重みと隣接状況が独立であるとすると，重みの大きい節点から逐次近似彩色を施していけば，特に隣接状況に比べ重みの影響が大きい場合は，$z(v)$ が小さくなることが期待できる．従って，最初に節点を重みの非増大順に整列 (sort) し，重みの大きい節点 v から出来るだけ小さい色番号 $k(v)$ を与えるものとする．

前記の様な逐次近似彩色と上界 $z(v)$ を求める操作を出来るだけ効率良く遂行するため，$z(v)$ を

$$z(v)=x(v)+y(v),$$

$$x(v)=\sum_{i=1}^{k(v)}\operatorname*{Max}_{u\in\Gamma(v)\cap C_i} w(u),$$

但し,$k(v)=1$ のときは，
$\Gamma(v)\cap C_1=\phi$ より，$x(v)=0$,

$$y(v)=w(v)+\sum_{i=k(v)+1}^{\tilde{\chi}}\operatorname*{Max}_{u\in\Gamma(v)\cap C_i} w(u),$$

但し,$k(v)=\tilde{\chi}$ のときは $y(v)=w(v)$.

の様に分割して考える．

すると，$x(v)$ を求めるためには v の色番号 $k(v)$ よりも大きい色番号をもつ節点は無影響であるので，出来る限り小さい色番号からによる逐次近似彩色の進行と並行して $x(v)$ を得ることが出来る．同様にして，$y(v)$ を求めるためには，逐次近似彩色が終了後，$k(v)$ よりも大きい色番号を持つ部分だけに関して計算を行えばよい．

ここで，C_i の中の節点の並び順も重みの非増大順になるようにし，節点間の隣接検査もその順に従うものとすれば,$x(v)$ を求める時，節点 $v\notin C_i$ に対して最初に見つかった隣接節点 $u\in C_i$ が直ちに $\operatorname{Max}_{u\in\Gamma(v)\cap C_i} w(u)$ なる値を持つ節点となる．次に $y(v)$ を求める際も同様であり，極めて単純な操作で済む．

4．無駄な彩色の削減

ある探索過程で最大重みクリーク Q が見出されているとする．その部分問題に対する節点集合 S に近似彩色の手続きを実行している途中において，節点集合 S 内で既に彩色操作が済んだ節点集合を $S^+=\{v_1,...,v_i\}$ とし，

$$\mathrm{cmw}(S^+)=\operatorname*{Max}_{u\in S^+}\{x(u)+w(u)\}$$

とおく．更に，まだ彩色操作を行っていない節点集合を $S''=S-S^+$ とおく．このとき，最後まで近似彩色の手続きを実行して得られる cmw を $\mathrm{cmw}(S)$ とすると

$$\mathrm{cmw}(S)\leq\mathrm{cmw}(S^+)+w(S'')$$

の不等式が成立する．従って，もし

$$w(Q)+\mathrm{cmw}(S^+)+w(S'')\leq w(Q_{max})$$

のようになった場合，

$$w(Q)+\operatorname*{Max}_{a\in S}\{x(a)+w(a)\}\leq w(Q_{max})$$

が導かれる．よって，この節点集合 $Q\cup S$ は $w(Q_{max})$ より重いクリークは含まない．そこで，これ以上の近似彩色の操作は無駄であり，この時点で近似彩色の操作を中止する．

5. アルゴリズム

```
procedure MWCL+ (G = (V, E), w)
  begin
    global Q_max := φ, Q := φ
    WEXTend(V)
    output Q_max
  end{ of MWCL+}

procedure WEXTend(V)
  begin
    WNELimin(V, y, w(Q_max) − w(Q))
    while V ≠ φ do
      v := the 1st vertex in V
      if w(Q) + y(v) > w(Q_max) then
        Q := Q ∪ {v}
        V' := V ∩ Γ(v)
        if V' ≠ φ then WEXTend(V')
        else if w(Q) > w(Q_max) then
          Q_max := Q fi
        fi
        Q := Q − {v}
      fi
      V := V − {v}
    od
  end {of WEXTend}
```

図 1. MWCL+

前章までの考察に基づいて,最大重みクリークを抽出する新しいアルゴリズムMWCL+(図.1)を提唱する. 逐次近似彩色を用いてクリーク重みの上界を得る手続きを,WNELimin(図.2) に示す. このとき,$x(v)+y(v)+w(Q) \leq w(Q_{max})$ のような v であれば, その節点 v は削除される. また, 節点の重みが一定 ($\Re = 0$) の場合はごく単純なものにおきかえるようにしている.

```
procedure WNELimin(V, y, l)
  begin
    χ̃ := 1, cmw := 0
    C_1 := φ, C_2 := φ
    sort V in non-increasing weight
    ℜ := Max_{v∈V} w(v) − Min_{u∈V} w(u)
    while V ≠ φ do
      if w(Q)+cmw+w(V) ≤ w(Q_max)
      then
        goto End fi
      v := the 1st vertex in V
      k := 1, x(v) := 0
      while C_k ∩ Γ(v) ≠ φ do
        u := the 1st vertex in C_k∩Γ(v)
        x(v) := x(v) + w(u)
        k := k + 1
      od
      if cmw < x(v) + w(v) then
        cmw := x(v) + w(v) fi
      if k > χ̃ then
        χ̃ := k, C_{k+1} := φ fi
      C_k := C_k ∪ {v}
      V := V − {v}
    od
    if ℜ > 0 then
      for k := 1 to χ̃ do
        while C_k ≠ φ do
          v := the 1st vertex in C_k
          y(v) := w(v)
          for h := k+1 to χ̃ do
            u := the 1st vertex in
            C_h ∩ Γ(v)
            y(v) := y(v) + w(u)
          od
          if x(v) + y(v) > l then
            V := V ∪ {v} fi
          C_k := C_k − {v}
        od
      od
    else
      V := C_χ̃ ∪ ··· ∪ C_2 ∪ C_1
      for all v ∈ V do
        y(v) := x(v) + w(v)
      od
    fi
    End:
  end {of WNELimin}
```

図 2. WNELimin

6. 計算機実験

前節に示したアルゴリズムの性能評価を計算機実験により行う. 対象とするグラフは, すべてランダムグラフとする. なお, 計

算機環境として,CPU PentiumII 266MHz Memory 128MB を搭載した AT 互換機に Linux をインストールしたものを使用し,アルゴリズムの実働化にはC言語を使用した.このときのプログラムのコンパイルは gcc -O2 で行なった.

節点数と枝の存在確率の異なる各グラフの実行結果を表 1.〜2. に示す. 表中の n, p とはそれぞれ節点数,枝の存在確率のことであり,測定値は5つのランダムグラフに対する実行結果の平均である. また,グラフ中の節点の重みは,文献 [8] などと同様に,正整数 1〜10 の一様乱数から与えている.

この結果,Babel[8] のアルゴリズムより探索分枝数は多いが,実行時間では高速であることが確認できた.

表 1. 平均実行時間 (単位 ms)

n	p	重み	MWCL+	Babel
100	0.3	44.8	2	22
	0.5	65.6	10	76
	0.7	101.8	40	368
200	0.3	52.2	10	156
	0.5	79.6	118	1,558
	0.7	125.4	3,096	27,267
300	0.3	59.8	54	716
	0.5	89.4	880	10,722
	0.7	142.6	70,118	649,824
400	0.3	62.4	142	2,354
	0.5	96.4	3,404	44,886
	0.7	155.0	542,842	5,756,054
500	0.3	66.2	342	5,580
	0.5	98.6	13,980	186,918
	0.7	164.4	4,417,614	45,407,002
600	0.3	67.6	762	12,288
	0.5	105.6	39,598	576,466
700	0.3	71.4	1,336	22,776
	0.5	109.0	93,712	1,395,206

表 2. 探索分枝数

n	p	重み	MWCL+	Babel
100	0.3	44.8	71	56
	0.5	65.6	190	79
	0.7	101.8	783	242
200	0.3	52.2	366	142
	0.5	79.6	2,434	1,285
	0.7	125.4	44,179	10,608
300	0.3	59.8	1,287	433
	0.5	89.4	15,059	5,833
	0.7	142.6	790,468	194,991
400	0.3	62.4	3,173	1,660
	0.5	96.4	47,026	19,686
	0.7	155.0	4,672,023	1,348,642
500	0.3	66.2	6,269	4,462
	0.5	98.6	182,725	85,459
	0.7	164.4	34,717,884	9,300,920
600	0.3	67.6	12,991	10,491
	0.5	105.6	466,430	229,652
700	0.3	71.4	20,054	17,212
	0.5	109.0	997,549	517,339

謝辞 討論,協力をいただいた本学若月光夫助手,元卒研生鄭戴勇氏 (現,三菱マテリアル) に感謝します. 本研究は文部省科学研究費の補助を受けている.

参考文献

[1] P. M. Pardalos and J. Xue: "The maximum clique problems," J . Global Optimization, vol. 4, pp.301-328 (1994).

[2] 富田悦次, 今松憲一, 木幡康宏, 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, " 信学論 (D-I), vol. J79- D-I, no.1, pp.1-8 (1996).

[3] 富田悦次, 平賀直仁, 若月光夫: "最大クリーク抽出アルゴリズムの効率化とその評価, " 情処研報, MPS24-1, pp.1-4 (1999).

[4] 秋山泰, 古谷立美: "対称相互結合型ニューラルネットを用いた大規模なRNAの二次構造予測," 信学技報 NC90-62, pp.57-64 (1991).

[5] 田中健夫, 若月光夫, 富田悦次: "最大重みクリークの近似抽出解法によるRNAの二次構造予測, " 信学会情報・システムソサイエティ大会, D-10 (1995).

[6] 若月光夫, 田中健夫, 富田悦次: "最大重みクリーク抽出アルゴリズムのRNAの二次構造予測への適用, " 情処 52 回大会,3U-9 (1996).

[7] 田中健夫, 若月光夫, 富田悦次: "組合せ最適化手法によるシュノードノット構造を含んだRNAの二次構造予測, " 情処研報, MPS10-4, pp.25-32 (1996).

[8] L. Babel: "A fast algorithm for the maximum weight clique problem," Computing. vol. 52, no.1,pp.31-38 (1994).

[9] 今松憲一, 富田悦次, 若月光夫: "近似彩色を用いた単純な最大重みクリーク抽出アルゴリズム, " 電通大紀要, vol. 8, no.1, pp.17-21 (1995).