アルゴリズム 20－3
(1991. 3. 15)

グラフをf辺彩色する近似アルゴリズム

中野眞一　藤井秀彦　西関隆夫

東北大学工学部情報工学科

グラフ $G=(V,E)$ の f彩色とは同じ色の辺が各点 vの接続辺中に高々 $f(v)$ 本しか存在しないようにグラフの全ての辺を彩色することである．グラフを高々 $\max_{v,w\in V}\left\lceil\frac{d(v)+p(vw)}{f(v)}\right\rceil$ 個の色を用いて f彩色する $O(|E|^2)$ 計算時間アルゴリズムは容易に得られる．ここで $d(v)$ は点 vの次数，$p(vw)$ は点 vと wを結ぶ多重辺の多重度である．本論文では同じ個数の色を用いて f彩色する $O(|E|\Delta_f\log|E|)$ 計算時間アルゴリズムおよび $O(|E|\sqrt{|E|\log|E|})$ 計算時間アルゴリズムを与える．

An Approximation Algorithms for f-Edge-Coloring Multigraphs

Shin-ichi Nakano　Hidehiko Fujii　Takao Nishizeki

Department of Information Engineering
Faculty of Engineering, Tohoku University
Sendai-shi 980, Japan

An f-coloring of a multigraph $G=(V,E)$ is a coloring of edges such that each color appears at each vertex v at most $f(v)$ times. One can easily obtain an algorithm which f-colors multigraphs with at most $\max_{v,w\in V}\left\lceil\frac{d(v)+p(vw)}{f(v)}\right\rceil$ colors and requires $O(|E|^2)$ time, where d(v) denotes the degree of vertex v, and $p(vw)$ the cardinality of the set of multiple edges joining vertices v and w. This paper presents two f-coloring algorithms which use the same number of colors and require $O(|E|\Delta_f\log|E|)$ or $O(|E|\sqrt{|E|\log|E|})$ time.

1. まえがき

本文では自己ループはないが多重辺(並列辺)はありうる多重グラフ $G=(V,E)$ を扱い，単にグラフと呼ぶ．ここで V は G の点集合であり，E は辺集合である．$d(v)$ は点 $v\in V$ の次数，$E(vw)$ は点 v と w を結ぶ多重辺の集合，$p(vw)=|E(vw)|$ はその多重度を表す．$\Delta=\max_{v\in V}d(v)$ とする．**点容量関数** f は V から自然数への任意の関数とする．$\Delta_f=\max_{v\in V}\lceil d(v)/f(v)\rceil$ とする．このとき任意の点 $v\in V$ に接続している同じ色の辺は高々 $f(v)$ 本しかないようにグラフ G のすべての辺を彩色することをグラフ G の f 彩色と呼ぶ．従来の辺彩色はすべての点 $v\in V$ について $f(v)=1$ なる場合の f 彩色に過ぎない．

グラフの f 彩色は並列処理等のスケジューリング問題に応用できる．類似したスケジューリング問題が文献(2)および(10)で扱われている．

グラフ G を f 彩色するのに必要な最小の色数を G の **f 彩色指数**といい，$\chi'_f(G)$ と書くことにする．グラフ G の通常の辺彩色に必要な最小の色数，いわゆる彩色指数 $\chi'(G)$ を求める問題は NP 困難であるので[9]，$\chi'_f(G)$ を求める問題も一般には NP 困難であり，$\chi'_f(G)$ を決定する効率のよいアルゴリズムはありそうにない[1]．

$\chi'_f(G)$ の上界として

$$\chi'_f(G)\le\max_{v,w\in V}\left\lceil\frac{d(v)+p(vw)}{f(v)}\right\rceil$$

が知られている[7]．これは通常の辺彩色に関する Vizing の定理[14]の一般化になっている．またグラフを上の上界が保証した個数の色を用いて f 彩色する $O(|E|^2)$ 時間アルゴリズムが上界の証明より得られる．本論文ではグラフを上の上界が保証した個数の色を用いて f 彩色する $O(|E|\Delta_f\log|E|)$ 時間アルゴリズムおよび $O(|E|\sqrt{|E|\log|E|})$ 時間アルゴリズムを与える．2つのアルゴリズムの記憶領域は $O(|V|\Delta_f+|E|)$ である．単純グラフを高々 $\Delta+1$ 色で辺彩色する $O(|E|\Delta\log|V|)$ 時間アルゴリズムおよび $O(|E|\sqrt{|V|\log|V|})$ 時間アルゴリズムが知られている[5]．2つのアルゴリズムの記憶領域は $O(|E|)$ である．本論文のアルゴリズムはこれを f 彩色に応用できるように工夫したものである．

2. 準備

本章ではまず用語の定義を与えてから，交互歩道のスイッチと扇のシフトという f 彩色の手法について説明する．

$v,w\in V$ のとき vw は v と w を結ぶ1本の辺を表す．上述の $\chi'_f(G)$ の上界を $u(G)$ とおく．すなわち

$$u(G)=\max_{v,w\in V}\left\lceil\frac{d(v)+p(vw)}{f(v)}\right\rceil$$

とする．本文のアルゴリズムはグラフ G を高々 $u(G)$ 個の色で f 彩色する．$u(G)$ 個の色の集合を U とする．色 $\alpha\in U$ で塗られている辺を α 辺と呼ぶ．点 v に接続している α 辺の本数を $d(v,\alpha)$ と書く．点 $v\in V$ の色 $\alpha\in U$ に関する**余裕** $m(v,\alpha)$ を

$$m(v,\alpha)=f(v)-d(v,\alpha)$$

と定める．

グラフ G が f 彩色されているとき，任意の点 $v\in V$ および色 $\alpha\in U$ について $m(v,\alpha)\ge 0$ である．$m(v,\alpha)\ge 1$ のとき点 v で色 α は**余裕色**であるという．点 v の余裕色の集合を $M(v)$ とする．すなわち $M(v)=\{\alpha\in U|m(v,\alpha)\ge 1\}$ とする．

余裕色について次の補題が成立する．

[補題 1]　任意のグラフ G の一部が $u(G)$ 個の色を使用して f 彩色されているとする．このとき点 $v\in V$ に接続する未彩色辺が a 本存在するなら任意の辺 $vw\in E$ において $\sum_{c\in U}m(v,c)\ge p(vw)+a$ である．

(証明) $u(G)$ の定義より $u(G)\cdot f(v)\ge d(v)+p(vw)$ である．よって $\sum_{c\in U}m(v,c)=u(G)\cdot f(v)-(d(v)-a)\ge p(vw)+a$ である．

(証明終)

補題1より任意の点 $v\in V$ において $\sum_{c\in U}m(v,c)\ge 1$ がいえる．

グラフ G の**歩道** W とは $v_0e_1v_1e_2v_2\cdots v_{k-1}e_kv_k$ の形をした辺点列である．但し各 i について $e_i\in E(v_{i-1}v_i)$ とし，W に同じ点は2度以上現れるかもしれないが，同じ辺は2度以上現れないものとする．v_0 を W の**始点**といい v_k を**終点**という．$v_0=v_k$ なる歩道を**閉路**という．W に含まれる辺数を $|W|$ と書き，W の**長さ**と呼ぶ．相異なる色 $\alpha,\beta\in U$ で交互に塗られた歩道 W のすべての α 辺を β で塗り，すべての β 辺を α で塗ることを W を**スイッチ**するという．W をスイッチしても f 彩色のままであるように **$\alpha\beta$ 交互歩道** W を次の様に定義する．

(a) W の辺は色 α と β で交互に塗られている．奇数番目の辺 e_i は β 辺であり，偶数番目の辺 e_i は α 辺である．
(b) (始点条件) $v_0\ne v_k$ のとき $m(v_0,\alpha)\ge 1$ である．$v_0=v_k$ かつ $|W|=k$ が奇数のとき $m(v_0,\alpha)\ge 2$ である．
(c) (終点条件) $v_k\ne v_0$ かつ $|W|$ が偶数のとき $m(v_k,\beta)\ge 1$ である．$v_k\ne v_0$ かつ $|W|$ が奇数のとき $m(v_k,\alpha)\ge 1$ である．

α,β で交互に塗られた偶数長の閉路は全て $\alpha\beta$ 交互歩道であることに注意しよう．v_0 を始点とする $\alpha\beta$ 交互歩道で偶数長の閉路でないものの一つを $W(\alpha,\beta,v_0)$ と書くことにする．$W(\alpha,\beta,v_0)$ をスイッチすると点の余裕は始点と終点のみで変化することに注意しよう．次の補題がいえる．

[補題 2]　グラフ G が f 彩色されているとき $m(v_0,\alpha)\ge 1$ かつ $m(v_0,\beta)=0$ ならば，$W(\alpha,\beta,v_0)$ が存在する．

(証明) $m(v_0,\alpha)\ge 1$ かつ $m(v_0,\beta)=0$ より $d(v_0,\beta)>d(v_0,\alpha)\ge 0$ なので v_0 に接続する β 辺が存在する．この β 辺を $e_1\in E(v_0v_1)$ とする．もし $m(v_1,\alpha)\ge 1$ ならば $v_0e_1v_1$ は $W(\alpha,\beta,v_0)$ である．もし $m(v_1,\alpha)=0$ ならば $d(v_1,\alpha)\ge d(v_1,\beta)$ であり，v_1 に接続する α 辺が存在する．この α 辺を $e_2\in E(v_1v_2)$ とする．以下同様にして同じ辺が2度以上現れないように α 辺と β 辺を交互に選ぶことを始点，終点条件が満たされるまで繰返す．特に始点 v_0 に α 辺で戻ってきたとき，すなわち偶数長の閉路を構成した場合は，$d(v_0,\beta)>d(v_0,\alpha)\ge 0$ より，v_0 に接続する β 辺でまだ $W(\alpha,\beta,v_0)$ に含まれていないものが存在する．その β 辺を $W(\alpha,\beta,v_0)$ に入れる．このようにすれば，偶数長の閉路でない $\alpha\beta$ 交互歩道すなわち $W(\alpha,\beta,v_0)$ を作ることができる．

(証明終)

次に彩色扇を定義する．辺 $e_0=wv_0$ を未彩色辺とする．点 w に接続する相異なる辺の列 $F=e_0e_1\cdots e_k$ を**彩色扇**と呼ぶ．但し各 i について $e_i\in E(wv_i)$ とし，次の条件(a)-(e)を満足する色 α と色の列 $\beta_0,\beta_1,\cdots,\beta_k$ が存在するものとする．w を F の**中心点**，v_i を F の**葉点**と呼ぶ．

(a) $\alpha\in M(w)$ である．
(b) $0\le i\le k-1$ なる各 i について $\alpha\notin M(v_i)$ である．
(c) $0\le i\le k$ なる各 i について $\beta_i\in M(v_i)$ である．
(d) $1\le i\le k$ なる各 i について辺 e_i は色 β_{i-1} で塗られている．
(e) 色 $\beta_0,\beta_1,\cdots,\beta_{k-1}$ は相異なる．

上の彩色扇を **α 彩色扇**または **$\alpha\beta_k$ 彩色扇**と呼ぶこともある．

点 $v_0,v_1,v_2,\cdots,v_k$ は相異なるとは限らないことに注意しよう．$0\le i\le k-1$ なる各辺 e_i を新たに色 β_i で塗り，辺 e_k を未彩色にすることを扇 F を**シフト**するという．条件(c),(d)により扇をシフトしても f 彩色は壊れない．特に未彩色辺1本だけで構成される彩色扇のシフトは何も行なわないことに等しい．

次に**極大な彩色扇** F を定義する．次の(a),(b)いずれかを満たすとき彩色扇は**極大**であるという．

(a) $m(w,\beta_k)+m(v_k,\beta_k)\geq 2$である．
(b) $\beta_i=\beta_k$なる$i(0\leq i\leq k-1)$が存在する．

次の補題が成立する．

[補題3] 任意のグラフGの一部が$u(G)$個の色を使用してf彩色されているとする．任意の未彩色辺$e_0=wv_0$および点wの任意の余裕色$\alpha\in U$に対しwを中心としe_0を含む極大なα彩色扇Fが存在する．
(証明) 補題1よりv_0には余裕色が存在する．v_0の任意の余裕色をβ_0とする．もし$m(v_0,\alpha)\geq 1$ならばv_0の余裕色をαと選び直すと，$F=e_0$はwを中心とし，色列αをもつ極大な$\alpha\alpha$彩色扇である．もし$m(w,\beta_0)+m(v_0,\beta_0)\geq 2$ならば$F=e_0$は$w$を中心とし，色列$\beta_0$をもつ極大な$\alpha\beta_0$彩色扇である．いずれでもなければ$m(w,\alpha)\geq 1$，$m(v_0,\alpha)=0$，$m(w,\beta_0)=0$，$m(v_0,\beta_0)=1$であり，$w$には$\beta_0$辺が接続している．この$\beta_0$辺を$e_1=wv_1$として彩色扇に付け加えて，$v_1$の余裕色を$\beta_1$とし，彩色扇の構成を続ける．ただし$v_0=v_1$のときは$\beta_1\neq\beta_0$のように選ぶ．補題1よりそのような色$\beta_1$が存在する．

いま一般に扇$F=e_0e_1\cdots e_k$が作られたとする．$v_i=v_k$なる各$0\leq i\leq k-1$に対して$\beta_i\neq\beta_k$となるように色β_kを選ぶこととする．$v_i=v_k$なる各$0\leq i\leq k-1$に対して$m(v_i,\beta_i)=1$が成立するので，補題1よりそのような色β_kが存在する．もし$m(v_k,\alpha)\geq 1$ならばv_kの余裕色をαと選び直すと，$F=e_0e_1\cdots e_k$はwを中心とし，色列$\beta_0,\beta_1,\cdots,\beta_{k-1},\alpha$をもつ極大な$\alpha\alpha$彩色扇である．もし$m(w,\beta_k)+m(v_k,\beta_k)\geq 2$または$\beta_i=\beta_k$なる$i(0\leq i\leq k-1)$が存在するならば$F=e_0e_1\cdots e_k$は$w$を中心とし，色列$\beta_0,\beta_1,\cdots,\beta_k$をもつ極大な$\alpha\beta_k$彩色扇であり，ここで彩色扇の構成を終了する．そうでないときは$m(w,\alpha)\geq 1$，$m(v_k,\alpha)=0$，$m(w,\beta_k)=0$，$m(v_k,\beta_k)=1$であり，wにはβ_k辺が接続している．このβ_k辺を$e_{k+1}=wv_{k+1}$として彩色扇に付け加えて，v_{k+1}の余裕色をβ_{k+1}とし，彩色扇の構成を続ける．

以上のようにして作られた彩色扇は極大なα彩色扇である．
(証明終)

3. アルゴリズム COLOR

本章ではアルゴリズム COLOR を与える．アルゴリズム COLOR は任意のグラフGを高々$u(G)$色使用して$O(|E|^2)$計算時間，$O(|V|\Delta_f+|E|)$記憶領域でf彩色する．アルゴリズム COLOR は4章，5章で与えるアルゴリズムの基本サブルーチンとしても使われる．

Procedure COLOR(G)
begin
  while Gに未彩色辺vwが存在する **do** RECOLOR(vw)
end

サブルーチン RECOLOR はGの1本の未彩色辺を彩色する．

Procedure RECOLOR(wv_0)
begin
  $\alpha\in M(w)$なる任意の色$\alpha\in U$を選ぶ;
  $F=e_0(=wv_0)e_1e_2\cdots e_k$は極大な$\alpha$彩色扇とする;
  Fの色列を$\beta_0,\beta_1,\beta_2\cdots,\beta_k$とする;
  if Fは$\alpha\alpha$彩色扇である { 場合1 }
  then begin
    Fをシフトする;
    e_kをαで塗る
  end
  else if $m(w,\beta_k)\geq 1$かつ$m(v_k,\beta_k)\geq 1$ { 場合2 }
  then begin
    Fをシフトする;
    e_kをβ_kで塗る
  end
  else if $m(v_k,\beta_k)\geq 2$ { 場合3 }
  then begin
    Fをシフトする;
    $W(\beta_k,\alpha,v_k)$をスイッチする;
    if Wがwで終っていなかった
    then e_kをαで塗る
    else e_kをβ_kで塗る
  end
  else ある$t(0\leq t\leq k-1)$について$\beta_t=\beta_k$である { 場合4 }
    if $W(\beta_k,\alpha,v_k)$はwおよびv_tで終了しない
    then $W(\beta_k,\alpha,v_k)$をスイッチし，Fをシフトし，e_kをαで塗る
    else if $W(\beta_k,\alpha,v_k)$がv_tで終了する
    then $W(\beta_k,\alpha,v_k)$をスイッチし，部分彩色扇$F'=e_0e_1\cdots e_t$をシフトし，e_tをαで塗る
    else { $W(\beta_k,\alpha,v_k)$がwで終了する }
      $W(\beta_k,\alpha,v_k)$をスイッチし，部分彩色扇$F'=e_0e_1\cdots e_t$をシフトし，e_tを$\beta_k=\beta_t$で塗る
end

場合4の$W(\beta_k,\alpha,v_k)$はFのβ_k辺$e_{t+1}=wv_{t+1}$を最終辺以外としては含まないように構成する．これには次の(a)，(b)を満足するように$W(\beta_k,\alpha,v_k)$を構成すればよい．
(a) もしα辺uv_{t+1}，$u\in V$，が選ばれたら，次にβ_k辺$v_{t+1}w$を選びWを終了する．
(b) もしα辺uw，$u\in V$，が選ばれたら，次にβ_k辺$e_{t+1}=wv_{t+1}$以外のβ_k辺を選びWの構成を続ける．

これらの条件により次のことがいえる．Wがwおよびv_tで終了しないときは，WはF中のβ_k辺wv_{t+1}を含まず，Wのスイッチは彩色扇Fを壊さない．Wがv_tで終了するときは，Wのスイッチ後部分彩色扇$F'=e_0e_1\cdots e_t$は，色列$\beta_0,\beta_1,\cdots,\beta_{t-1},\alpha$をもつ$\alpha\alpha$彩色扇になる．$W$が$w$で終了するときは，$W$は$F$中の$\beta_k$辺$v_{t+1}w$を最終辺として含む．このとき$W$のスイッチは部分彩色扇$F'=e_0e_1\cdots e_t$を壊さない．以上により RECOLOR の場合4は正しく処理される．

[定理1] アルゴリズム COLOR は任意のグラフGを高々$u(G)$色使用して$O(|E|^2)$計算時間，$O(|V|\Delta_f+|E|)$記憶領域でf彩色する．
(証明) データ構造は以下のものを用いる．

(a) 点隣接リスト: 各点に接続している辺をリスト構造を用いてつなぐ．$O(|V|+|E|)$の記憶領域が必要である．
(b) 同色辺リスト: 同じ色で塗られている辺を双方向リストでつなぐ．未彩色辺も双方向リストでつなぐ．$O(|E|)$の記憶領域が必要である．
(c) 点色辺配列: 各点に接続する同じ色の辺を双方向リストでつなぐ．また各点に接続する未彩色辺も双方向リストでつなぐ．これらのリストへのポインタを$|V|\times(|U|+1)$の2次元配列に入れる．$O(|V|\Delta_f+|E|)$の記憶領域が必要である．
(d) 点余裕配列: $m(v,\alpha)$の値を$|V|\times|U|$の2次元配列に入れる．$O(|V|\Delta_f)$の記憶領域が必要である．
(e) 点余裕リスト: 各点$v\in V$について$m(v,\beta)\geq 1$なる色$\beta\in U$をすべて双方向リストでつなぐ．これにより$M(v)$を表現する．$O(|V|\Delta_f)$の記憶領域が必要である．点余裕リストの各データは対応する点余裕配列中のデータと互いに直接アクセスできるようにポインタでつないでおく．

色αで塗られた辺vwはvとwの2つの点隣接リストおよび色αの同色辺リストおよび点色辺配列中のvに接続するα辺リストとwに接続するα辺リストに入っている．

これら5つの同一辺に関するデータを直接互いにアクセスできるようにするため，$|E|\times 5$の2次元配列を用意し，各データへのポインタを記憶する．

以上全てに要する記憶領域は$O(|V|\Delta_f+|E|)$である．

これらのデータ構造を用いることにより以下がいえる．

(a) 極大な彩色扇は $O(|U|)$ 時間で作れる．
(b) 彩色扇のシフトは $O(|U|)$ 時間で実行できる．
(c) 交互歩道は $O(|V|+|E|)$ 時間で作れる．
(d) 交互歩道のスイッチは $O(|V|+|E|)$ 時間で実行できる．

1本の辺を彩色するのに上の (a)-(d) は高々定数回しか実行しない．よって1本の辺は $O(|V|+|E|)$ 時間で彩色できる．

よってアルゴリズム全体の計算時間は $O(|E|(|V|+|E|))$ すなわち $O(|E|^2)$ である． (証明終)

4. アルゴリズム PARALLEL-COLOR

本章では任意のグラフ G を高々 $u(G)$ 色で f彩色するアルゴリズム PARALLEL-COLOR を示す．このアルゴリズムの計算時間は $O(|E|\Delta_f \log|E|)$，記憶領域は $O(|V|\Delta_f+|E|)$ である．

アルゴリズム COLOR は1本ずつ未彩色辺を彩色し，1本の交互歩道の構成とスイッチに $O(|V|+|E|)$ 時間かかった．アルゴリズム PARALLEL-COLOR では多くの未彩色辺を並列的に彩色しており，複数本の交互歩道の構成とスイッチが $O(|V|+|E|)$ 時間ですむ．

本アルゴリズムは$\alpha\beta$交互歩道を以下のように構成する．G のすべてのα辺とβ辺から誘導される部分グラフを $G(\alpha,\beta)$ と書くことにする．はじめに $G(\alpha,\beta)$ に点 x を加え，$v \in V$なる各点との間に $m(v,\alpha)$ 本のα辺と $m(v,\beta)$ 本のβ辺を加えたグラフ G' を作る．$f(x)=\sum_{v\in V}\{m(v,\alpha)+m(v,\beta)\}$ とする．G' の $v \neq x$ なる各点にはα辺とβ辺が $f(v)$ 本ずつ接続する．つぎに G' を互いに辺素な$\alpha\beta$交互歩道と$\beta\alpha$交互歩道の集合に分割する．ただし，すべての交互歩道は x を始点かつ終点とするものとする．最後に上記の各交互歩道から x に接続する辺をすべて削除すると (交互歩道は分割されるかもしれない) $\alpha\beta$交互歩道と$\beta\alpha$交互歩道の集合が残る．

このようにして $G(\alpha,\beta)$ を互いに辺素な$\alpha\beta$交互歩道と$\beta\alpha$交互歩道の集合に分割する．こうしてできた交互歩道の集合は以下を満足する．

(a) $v \in V$が m_1本の $W(\alpha,\beta,v)$ の始点，かつ m_2本の奇数長の $W(\alpha,\beta,w)$，$w \in V$，の終点，かつ m_3本の偶数長の $W(\beta,\alpha,w)$，$w \in V$，の終点ならば $m(v,\alpha)=m_1+m_2+m_3$ である．すなわち各交互歩道の端点でその交互歩道が独占できる色αの余裕がある．よって任意本数の$\alpha\beta$交互歩道をスイッチしても他の$\alpha\beta$交互歩道を破壊しない．

(b) $v \in V$において $d(v,\beta) \geq d(v,\alpha)$ ならば少なくとも $d(v,\beta)-d(v,\alpha)$ 本の $W(\alpha,\beta,v)$ が存在する．ただし，$W(\alpha,\beta,v)$ が奇閉路のときは (Wの辺列を逆順にたどる $W'(\alpha,\beta,v)$ も存在するので) これを2本と数える．

本章のアルゴリズムはアルゴリズム COLOR を並列的に構成したものである．はじめに複数の彩色扇を作り，次にそれぞれの彩色扇に必要な交互歩道を作り，最後にそれぞれの彩色扇を彩色する，(いくつかの交互歩道はスイッチされるかもしれない．) というようにアルゴリズムは進む．よって，彩色扇を彩色するときは他の彩色扇や交互歩道を破壊しないように互いに独立に彩色したい．

交互歩道に関して彩色の独立性を考えてみよう．まず (a) より上記のように構成した交互歩道はスイッチしても他の交互歩道を破壊しない．また (b) より $m(v,\beta)=0$ のとき，少なくとも $m(v,\alpha)$ 本の $W(\alpha,\beta,v)$ が存在する．これは各点を始点とした交互歩道が本アルゴリズムで必要な本数存在することを意味する．各彩色扇は必要なだけの交互歩道を独占するように本アルゴリズムは進む．ただし，交互歩道 Wがある彩色扇に独占されたとき，Wの辺列を逆順にたどる交互歩道 W'が別の彩色扇に独占されることの無いようにしたい．本アルゴリズムではこのようなとき一方の部分の彩色を断念する．

本章の主アルゴリズム PARALLEL-COLOR を次に示し，以下説明する．

Procedure PARALLEL-COLOR(G)
begin
1. **while** 未彩色辺がある **do**
2. **for** 各色$\alpha \in U$ **do**
  begin {3-9 行はαステージ}
  3. 両端点がα余裕な未彩色辺がある限り色αで塗る;
  4. TWO-COLOR(α);
  5. MAKE-S(α);
  6. PRESUBSTAGE(α);
  7. $\beta \neq \alpha$なる各色に適当に順番を付け$\beta_1,\beta_2,\cdots,\beta_{u(G)-1}$とする;
  8. **for** i=1 to $u(G)-1$ **do**
    9. SUBSTAGE($\alpha\beta_i$)
  end
end

まずアルゴリズムの概要を説明する．3-9 行はαステージ，9 行は$\alpha\beta_i$サブステージと呼ぶ．3,4 行で比較的容易に彩色できる未彩色辺を彩色する．5 行で互いに点素なα彩色扇の集合を作る．6 行でα彩色扇の集合を各サブステージで扱う部分集合に分割する．9 行でその部分集合中の互いに点素な彩色扇に含まれる未彩色辺を並列的に彩色する．後に示すように，2-9 行を1回実行すると未彩色辺の少なくとも 1/44 は彩色される．3-9 行は $O(|E|)$ 計算時間かかるので，アルゴリズム全体では $O(|E|\Delta_f \log|E|)$ 計算時間かかる．

以下に各サブルーチンの詳細を述べる．まずサブルーチン TWO-COLOR を説明する．TWO-COLOR ではα過余裕未彩色辺を彩色する．**α過余裕未彩色辺** wvとは $m(w,\alpha) \geq 2$ なる未彩色辺である．(α過余裕未彩色辺 vwでは $m(v,\alpha) \geq 2$ とする．) いま PARALLEL-COLOR の3行目を実行したので全ての未彩色辺 wvにおいて $m(w,\alpha) \geq 1$ ならば $m(v,\alpha)=0$ である．よって $m(v,\beta) \geq 1$ なる色$\beta \neq \alpha$が存在する．このときα過余裕未彩色辺 wvを彩色するには以下のようにする．もし $m(w,\beta) \geq 1$ ならば wvをβで塗る．そうでないときは $W(\alpha,\beta,w)$ をスイッチする．もし Wが vで終了しなければ wvをβで塗り，Wが vで終了すれば wvをαで塗る．(たとえ Wが奇閉路であってもスイッチの後 wvをβで塗れることに注意しよう．)

TWO-COLOR は多くのα過余裕未彩色辺を並列的に彩色する．TWO-COLOR が扱うα過余裕未彩色辺の集合を Rとする．

まずα過余裕未彩色辺 wvの端点 wでの色αの2つの余裕と端点 vでの任意の色βの1つの余裕は wvに独占させる．このようにして点 wに未彩色辺が p 本接続しているならば，そのうちの $\min\{\lfloor m(w,\alpha)/2\rfloor,p\}$ 本をα過余裕未彩色辺として集合 Rに入れる．また$\alpha \notin M(v)$ なる点 vが p'本の R中のα過余裕未彩色辺の端点になっているならば$\sum_{c\in U} m(v,c) \geq p'$であるので，$v$に接続する各$\alpha$過余裕未彩色辺について独占的に点 vのある余裕色β_iの1つの余裕を割当てることにする．このとき色β_iを割当てられたα過余裕未彩色辺の集合を $R(\beta_i)$ とする．すなわち $R(\beta_i)$，$1 \leq i \leq u(G)-1$，は Rの分割である．

次に各 $R(\beta_i)$ について $wv \in R(\beta_i)$ かつ $m(w,\beta_i) \geq 1$ なるα過余裕未彩色辺をβ_iで塗り Rおよび $R(\beta_i)$ から取り除く．

最後に各 $R(\beta_i)$ について以下のことを行なう．先に説明した方法で$\alpha\beta_i$交互歩道の集合を作る．任意の $wv \in R(\beta_i)$ を Rおよび $R(\beta_i)$ から取り除く．そして2本の $W(\alpha,\beta_i,w)$ と1本の $W(\beta_i,\alpha,v)$ を選び wvに排他的に独占させる．この3つの交互歩道がスイッチされなければ wvはα過余裕未彩色辺であり続ける．(ただし $W(\alpha,\beta_i,w)$ が奇閉路なら Wの辺列を逆順にたどる交互歩道も考えてこれを2本と数える．また少なくとも一方の $W(\alpha,\beta_i,w)$ の終点が vなら Wの辺列を逆順にたどる交互歩道を $W(\beta_i,\alpha,v)$ とみなし，他の $W(\beta_i,\alpha,v)$ は独占しないとする．)

そこで $W(\alpha,\beta,w)$ または $W(\beta,\alpha,v)$ の辺列を逆にたどる W' がスイッチされないことを保証したい．独占した交互歩道 W の終点条件が $m(v_k,\alpha) \geq 1$ のとき $v_kv' \in R$なる辺が存在するならば (複数あるときは任意の1本の) v_kv' を Rおよび対応する $R(\beta_j)$ から削除する．同様に終点条件 $m(v_k,\beta_i) \geq 1$ なる Wを独占したとき $w'v_k \in R$なる辺が存在するならば (複数あるときは任意の1本の) $w'v_k$を Rおよび $R(\beta_i)$ から削除する．3本の交互歩道を独占した wvは集合 $R'(\beta_i)$ および集合 R'に入れる．以

上を $R(\beta_i)$ が空になるまで繰返す．最後に $R'(\beta_i)$ 中のα過余裕未彩色辺をそれぞれ独立に彩色する．

すなわち 1 本のα過余裕未彩色辺の彩色のために 3 本の交互歩道を独占し，高々3 本のα過余裕未彩色辺の彩色を断念する．よって TWO-COLOR では少なくとも $|R|/4$ 本を彩色する．

Procedure TWO-COLOR(α)
begin
4.1 $R := \phi$;
4.2 **for** $m(w,\alpha) \geq 2$ なる各点 $w \in V$ **do**
4.3 wに接続する未彩色辺が p 本あるとき，そのうちの $\min\{\lfloor m(w,\alpha)/2 \rfloor, p\}$ 本を Rに入れる;
4.4 **for** 各辺 $e = wv \in R$ **do**
4.5 $\beta_i \in M(v)$ なる色$\beta_i \in U$をひとつ選び，vwを $R(\beta_i)$ に入れる;
4.6 **for** $\beta_i \neq \alpha$なる各色$\beta_i \in U$ **do**
begin
4.7 **for** $wv \in R(\beta_i)$ **do**
4.8 **if** $m(w,\beta_i) \geq 1$ **then** wvをβ_iで塗り，wvを $R(\beta_i)$ から取り除く;
4.9 $\alpha\beta_i$交互歩道の集合を作る;
4.10 **for** $wv \in R(\beta_i)$ **do**
4.11 2 本の $W(\alpha,\beta_i,w)$ と 1 本の $W(\beta_i,\alpha,v)$ を独占し，各 Wの終点に接続するα過余裕未彩色辺を必要に応じて取り除く;
4.12 **for** $wv \in R(\alpha\beta_i)$ **do**
begin
4.13 $W(\alpha,\beta_i,w)$ をスイッチする;
4.14 **if** Wは vで終了しなかった **then** wvをβで塗る
4.15 **else** wvをαで塗る
end
end
end

次にサブルーチン MAKE-S を説明する．まずα未彩色扇を定義する．点 wに接続する相異なる 2 本以上の未彩色辺の列 $F = e_0 e_1 \cdots e_k$をα未彩色扇と呼ぶ．但し各 i について $e_i \in E(wv_i)$ とし，次の条件 (a)-(c) を満足する色α が存在するものとする．wを Fの中心点，v_iを Fの葉点と呼ぶ．βが中心点 wの余裕色である未彩色扇を$\alpha\beta$未彩色扇と呼ぶ．

(a) $0 \leq i \leq k$なる各 i について$\alpha \in M(v_i)$ である．
(b) $\alpha \notin M(w)$．
(c) $v_0, v_1, \cdots, v_k$は相異なる．

サブルーチン MAKE-S は互いに点素な彩色扇と未彩色扇を要素とする集合 Sを作る．任意の彩色扇のシフトは互いに点素な彩色扇および未彩色扇を破壊しないことに注意しよう．

Procedure MAKE-S(α)
begin
5.1 $S := \phi$;
5.2 **for** $\alpha \in M(w)$ かつ wは S中の扇に属していない かつ wには未彩色辺 e_0が接続している各 $w \in V$について **do**
begin
5.3 $F = e_0 e_1 \cdots e_k$は極大α彩色扇とし，$e_i \in E(wv_i)$ とする;
5.4 **if** Fは Sの全ての扇と点素である
5.5 **then** $S := S \cup \{F\}$
else begin
5.6 Sの扇に属する点 v_iを i が最小となるように選ぶ;
v_iは $F' \in S$に属するとする;
w'は F'の中心点とする;
5.7 **if** F'は彩色扇である
then begin {F'はα彩色扇であり，v_iは F'の葉点}
5.8 Fの部分扇 $e_0 e_1 \cdots e_i$をシフトする;
5.9 F'の部分扇 $e'_0 e'_1 \cdots e'_j$，$e'_j \in E(w'v_i)$ をシフトする;
5.10 F''は 2 本の未彩色辺 e_iと e'_jからなる未彩色扇とする
;
5.11 $S := S \cup \{F''\} - \{F'\}$
end
else begin {F'はα未彩色扇．v_iは F'の中心点}
5.12 Fの部分彩色扇 $e_0 e_1 \cdots e_i$をシフトする;
5.13 F''は F'に未彩色辺 e_iを加えたものとする;
5.14 $S := S \cup \{F''\} - \{F'\}$
end
end
end
end

次のサブルーチン PRESUBSTAGE は集合 S中のシフトのみで彩色できる彩色扇 (RECOLOR の場合 1,2 のとき) を彩色し，これ以外の扇 (未彩色扇と RECOLOR 場合 3,4 の彩色扇) を集合 $S(\beta_j)$，$1 \leq j \leq |U|-1$ に分割する．

Procedure PRESUBSTAGE(α)
begin
6.1 **for** 各扇 $F \in S$ **do**
begin { wを $F = e_0 e_1 \cdots e_k$の中心とする．}
6.2 **if** Fはシフトのみで彩色可能なα彩色扇
6.3 **then** RECOLOR(F)
6.4 **else if** Fはα彩色扇
6.5 **then** $\beta_i \in M(v_k)$ なる色$\beta_i \in U$を選び，Fを $S(\beta_i)$ に入れる { Fは$\alpha\beta_i$彩色扇とする．}
else {Fはα未彩色扇};
6.6 $\beta_i \in M(w)$ なる最小の i を選び，Fを $S(\beta_i)$ に入れる
end
end

次に SUBSTAGE で使用するサブルーチン URECOLOR を示す．URECOLOR は未彩色扇の辺を 1 本彩色する．

Procedure URECOLOR(F) { Fは$\alpha\beta$未彩色扇であり，$F = e_0\ e_1 \cdots e_k$とする．すなわち $0 \leq i \leq k$なる各 i について $e_i \in E(wv_i)$ とすると，$\alpha \in M(v_i)$ かつ$\beta \in M(w)$ である．}
begin
$W(\beta,\alpha,w)$ をスイッチする;
if Wが x_0で終了しなかったか，または奇数長である
then e_0を色αで塗る
else e_kを色αで塗り，Fから辺 e_0を削除する
end

$\alpha\beta_i$交互歩道の終点が$\alpha\beta_i$未彩色扇 F' の中心点 wであるとき，これをスイッチすると$\alpha \in M(w)$ となり，F'はα未彩色扇ではなくなってしまう．SUBSTAGE で使用するサブルーチン REMOVE はこのように破壊されたα未彩色扇を修復するものである．

Procedure REMOVE(F) { Fは$\alpha\beta$彩色扇または$\alpha\beta$未彩色扇とする }
begin
if Fの彩色のためにスイッチした$\alpha\beta$交互歩道の終点が$\alpha\beta$未彩色扇 $F' \in S(\beta)$ の中心点である
then { $F' = e_0 e_1 \cdots e_k$とする }
e_kをαで塗る;
end
end

$j > i$，$F'' \in S(\beta_j)$ なる$\alpha\beta_j$彩色扇 F'' の中心点 wは$\alpha \notin M(w)$ $\beta_i \notin M(w)$ であり，$\alpha\beta_i$交互歩道の終点とはならないことに注意しよう．

最後に SUBSTAGE を説明する．SUBSTAGE($\alpha\beta_i$) は $S(\beta_i)$ 中の彩色扇に含まれる未彩色辺と未彩色扇の一部の未彩色辺を彩色する．塗り残した部分を持つ未彩色扇は適当な $S(\beta_j)$, $j > i$ に加えられ，後の SUBSTAGE($\alpha\beta_j$) で再び彩色されるようにアルゴリズムは進む．

いま S中の扇は互いに点素である．しかしある扇の彩色は交互歩道のスイッチにより任意個数の扇を破壊するかもしれない．そこで交互歩道は扇を構成するどの辺も最終辺以外には含まないようにしたい．このために G' 上で$\alpha\beta$交互歩道をつくるとき以下の様にする．β辺 $e = wv_i$を含む任意の彩色扇 F' において F' に含まれるβ辺は 1 本だけである．$m(w,\beta) = 0$，$m(w,\alpha) \geq 1$ である．G' 上で$\alpha\beta$交互歩道を構成するときβ辺 $e = wv_i$と任意のα辺 xw が連続するようにする．α辺 xwは後に除去されるのでβ辺 $e = wv_i$は必ず交互歩道の端の辺になる．本アルゴリズムの交互歩道の選びかたよりβ辺 $e = wv_i$は必ず交互歩道の最終辺になる．

SUBSTAGE では互いに独立に彩色できる部分のみ彩色を進め，いくつかの部分の彩色は断念する．1 本の未彩色辺の彩色に際して彩色を断念する未彩色辺は高々10 本である．

SUBSTAGE の詳細を説明する．$S(\beta_i)$ 中の各扇は以下の$\alpha\beta_i$交互歩道を独占する．

α未彩色扇 $F = wv_0, \cdots, wv_k$では $0 \leq j \leq \min\{k, m(w,\beta_i) - 1\}$ なる各 jについて 1 本ずつ $W(\alpha,\beta_i,v_j)$ と $W(\beta_i,\alpha,w)$ を独占する．RECOLOR の場合 3 のα彩色扇 $F = wv_0, \cdots, wv_i$では 2 本の $W(\beta_i,\alpha,v_i)$ と 1 本の $W(\alpha,\beta_i,w)$ を独占する．RECOLOR の場合 4 のα彩色扇 $F = wv_0, \cdots, wv_t, \cdots, wv_i$では 1 本の $W(\beta_i,\alpha,v_t)$ と 1 本の $W(\beta_i,\alpha,v_i)$ と 1 本の $W(\alpha,\beta_i,w)$ を独占する．

ただし Fの独占した交互歩道 Wの終点が Fに含まれかつその Wを逆にたどる交互歩道 W' が独占すべき交互歩道であるときはこれを 2 本の交互歩道とみなす．たとえば RECOLOR の場合 3 のα彩色扇 $F = wv_0, \cdots, wv_i$では 1 本の奇数長の閉路である $W(\beta_i,\alpha,w)$ と 1 本の $W(\alpha,\beta_i,w)$ を独占するだけとする．

いま交互歩道のスイッチは他の交互歩道を破壊しない．しかし交互歩道のスイッチは終点を含む他の扇をひとつ破壊するかもしれない．よって，上で作成した扇 Fの各$\alpha\beta_i$交互歩道 Wの終点 u により彩色を断念する辺を次の様に決める．

まず u が Sのどの扇にも含まれていないならば，なにもしない．u が Sの彩色扇 F' に含まれているならば F' の彩色を断念する．すなわち F' を Sおよび対応する $S(\beta_i)$ から取り除く．u が Sの 2 本の辺で構成される未彩色扇 F' に含まれているならば F' の彩色を断念する．u が Sの未彩色扇 $F' = \cdots, wv', \cdots$の葉点 v' ならば F' から辺 wv'を取り除く．以上でないとき u は 3 本以上の辺で構成される$\alpha\beta_i$未彩色扇 $F' = e_0, \cdots$の中心点である．このときは F' から辺 e_0を取り除く．そうでないとき F' の中心点が既に p 本の$\alpha\beta_i$交互歩道の終点であって，$p = m(u,\beta_i)$ なら，$\beta_j \in M(u)$ かつ $j > i$ なる最小の jを選び，F' を $S(\beta_j)$ に加え，後の SUBSTAGE($\alpha\beta_j$) で再び彩色されるようにする．

次に SUBSTAGE を示す．

Procedure SUBSTAGE($\alpha\beta_i$)
begin
9.1 $\alpha\beta_i$交互歩道の集合を作る;
9.2 **for** 各扇 $F \in S(\beta_i)$ **do** 必要な交互歩道を独占し，彩色を断念する辺を取り除く;
9.3 **for** 各扇 $F \in S(\beta_i)$ **do**
  begin
  9.4 $S(\beta_i) := S(\beta_i) - \{F\}$;
  9.5 **if** Fは未彩色扇である
    then begin
    9.6 **while** $m(w,\beta_i) \geq 1$ かつ Fは 2 本以上の未彩色辺を持つ **do**
      begin
      9.7 URECOLOR(F);
      9.8 REMOVE(F)
      end
    9.9 **if** Fは 2 本以上の未彩色辺をもつ
    9.10 **then** $\beta_j \in M(w)$ なる最小の $j > i$ を選び，$S(\beta_j) := S(\beta_j) \cup \{F\}$
    end
    else begin $\{F$は c-扇である．$\}$
    9.11 RECOLOR(F);
    9.12 REMOVE(F)
    end
  end
end

[定理 2] アルゴリズム PARALLEL-COLOR は任意のグラフ G を高々$u(G)$ 色使用し，$O(|E|\Delta_f \log|E|)$ 計算時間，$O(|V|\Delta_f + |E|)$ 記憶領域で f彩色する．

(証明) データ構造は COLOR と同様である．計算時間について考察する．

まず 1 回のαステージ (3–9 行) の計算時間を考察する．

TWO-COLOR で交互歩道の集合は $O(|E|)$ で作れる．各交互歩道は高々1 回参照される．

SUBSTAGE の交互歩道の参照およびスイッチに関する計算時間は交互歩道の辺数の定数倍時間である．α辺 1 本だけで構成される交互歩道に関する計算時間は新たに塗られた辺 1 本あたり定数時間なのでαステージで合計 $O(|E|)$ である．これ以外の交互歩道，すなわち少なくともβ辺を 1 本含む交互歩道に関する計算時間はβ辺数を E_β とすると $O(E_\beta)$ であり，αステージで合計 $O(\sum_{\beta\in U} E_\beta)$ すなわち $O(|E|)$ である．

Sの全ての扇は点素である．扇の構成およびシフトに関する操作は扇の辺数の定数倍時間であり，αステージで合計 $O(|E|)$ 時間である．

以上によりαステージの計算時間は $O(|E|)$ である．

$|U|$ は $O(\Delta_f)$ 色である．よって 2–9 行の計算時間は $O(|E|\Delta_f)$ である．

最後に 1 行目の while の繰返し回数を見積もる．$w \in V$に接続する未彩色辺が p 本あるとする．補題 1 より任意の辺 $wv \in E(wv)$ において$\sum_{c\in U} m(w,c) \geq p(wv) + p$ である．すなわち各点 wで wに接続する各未彩色辺に対応した余裕があり，各未彩色辺はいずれかのステージで扱われる．ただし $m(v,\beta) \geq 2$ のように余裕の重複があるとき TWO-COLOR により半分の未彩色辺が捨てられるかもしれない．また wで辺 wvのために用意した余裕がβであったとき，βステージが開始する前に交互歩道のスイッチまたは未彩色辺の彩色または扇のシフトによって w が辺 wvのために用意した余裕βが失われるかもしれない．

中心点 wの彩色扇のシフトによって，葉点 vの余裕は高々 $p(wv)$ 個変化する．すなわち p 個の余裕は保存される．またシフトによって葉点以外の点の余裕は変化しない．交互歩道のスイッチにより余裕の変化は始点と終点のみで生じる．すなわち交互歩道の終点 wにおいて wが未彩色辺 wvのために用意した余裕βが失われるかもしれない．また未彩色辺の彩色によっても一方の端点で他の未彩色辺のために用意した余裕が失われるかもしれない．

以上より 2–9 行を 1 回実行すると少なくとも未彩色辺の 1/4 はいずれかのステージで扱われる．(未彩色辺 vx, vwが存在し，$m(v,\alpha) = 2$ のときα過余裕辺 vwが $W = vv_1, \cdots, v_{k-1}y$のスイッチの後に色$\beta$で塗られたとしよう．このとき vwの彩色のため vx および vに接続するβステージで扱われるべき未彩色辺および y に接続するβステージで扱われるべき未彩色辺が 2–9 行を 1 回実行したとき，どのステージでも扱われない．)

また 2—9 行を実行すると扱った未彩色辺の少なくとも 1/11 を塗る．(3 本の未彩色辺 vw, vx, vyで構成される未彩色扇 Fが存在し，$m(v,\beta_1) = 3$ のとき，Fは 1 本の vを終点とする $W(\alpha,\beta,w)$，2 本の $W(\beta,\alpha,v)$，1 本の $W(\alpha,\beta,x)$，1 本の $W(\alpha,\beta,y)$ を独占したとする．それぞれ $W_1, W_2, \cdots, W_5$と呼ぼう．$W_2, \cdots, W_5$ の終点がそれぞれ 2 本の辺で構成される未彩色扇に含まれるとき SUBSTAGE の 9.2 行でこれらの 8 本の辺の彩色を断念する．URECOLOR では W_1の終点が vであるので W_1のスイッチ後 vy をαで塗り，vwの彩色は断念する．そして SUBSTAGE の 9.8 行で 1 本の未彩色辺 vx で構成される Fは捨てられる．これが最悪の場合である．)

よって 1 行目の **while** は $O(\log|E|)$ 回繰返される．

よってアルゴリズム全体の計算時間は $O(|E|\Delta_f \log|E|)$ である. (証明終)

5. アルゴリズム EULER-COLOR

本章では任意のグラフ G を高々$u(G)$ 色で f彩色するアルゴリズム EULER-COLOR を示す. このアルゴリズムの計算時間は $O(|E|\sqrt{|E|\log|E|})$, 計算領域は $O(|V|\Delta_f+|E|)$ である.

$G=(V,E)$ からオイラー分割を使ってつぎのように 2 つの部分グラフ $G_a=(V_a,E_a)$, $G_b=(V_b,E_b)$ をつくる手続きを EULER-P と呼ぶ.

まず $V_a=V_b=V$とする. G に多重辺対がある限り除去し, G_aと G_bに 1 本ずつ加える. こうしてできた単純グラフを G' とする. 次に G' の辺集合を互いに辺素な歩道に分割する. ただしどの閉路も他の歩道と交差せず, 奇数次数の点は丁度 1 つの歩道の端点になるようにする. 最後に各歩道の辺を交互に G_a, G_b に加える.

G_aの点 $v\in V_a$の次数を $d_a(v)$ とする. G_bの点 $v\in V_b$の次数を $d_b(v)$ とする. G_aの多重辺 $E(vw)\in E_a$の多重度を $p_a(vw)$ とする. G_bの多重辺 $E(vw)\in E_b$の多重度を $p_b(vw)$ とする.

以下の式が成立する. 奇閉路の始終点に注意しよう.

$$\lfloor(d(v)-1)/2\rfloor \le d_a(v) \le \lceil(d(v)+1)/2\rceil$$

$$\lfloor(d(v)-1)/2\rfloor \le d_b(v) \le \lceil(d(v)+1)/2\rceil$$

$$\lfloor p(vw)/2\rfloor \le p_a(vw) \le \lceil p(vw)/2\rceil$$

$$\lfloor p(vw)/2\rfloor \le p_b(vw) \le \lceil p(vw)/2\rceil$$

$$\lceil|E|/2\rceil = |E_a|$$

$$\lfloor|E|/2\rfloor = |E_b|$$

以上より

$$u(G_1)+u(G_2) \le u(G)+3$$

である.

任意のグラフ G を高々$u(G)$ 色で f彩色するアルゴリズム EULER-COLOR を示す. 定数 $c=\log\Delta_f\sqrt{(\log|E|)/|E|}$ とする.

Procedure EULER-COLOR(G)
begin
1. **if** G は EULER-COLOR により c 回以上分割された
2. **then** PARALLEL-COLOR(G)
   else begin
   3. EULER-P(G); $\{G_a$と G_bに G を分割する $\}$
   4. EULER-COLOR(G_a);
   5. EULER-COLOR(G_b);
   6. **if** G は $u(G)$ 以上の色を使っている
   7. **then** 辺数が最小の色を選び, その色で塗られている辺をすべて未彩色辺にする. G の使用する色が高々$u(G)$ 個になるまで (高々3 回) これを繰返す
   8. **while** G に未彩色辺 vwが存在する **do**
      9. RECOLOR(vw)
   end

end

[定理 3] アルゴリズム EULER-COLOR(G) は任意のグラフ G を高々$u(G)$ 色使用して $O(|E|\sqrt{|E|\log|E|})$ 計算時間, $O(|V|\Delta_f+|E|)$ 記憶領域で f彩色する.

(証明) PARALLEL-COLOR と同様のデータ構造を用いる. 記憶領域は $O(|V|\Delta_f+|E|)$ である.

グラフ G を i 回オイラー分割して得られた部分グラフ $G_i=(V_i,E_i)$ での点 $v\in V_i$の次数を $d(v,G_i)$ とする. 便宜的に $E_0=E$ と定める. 最大次数$\Delta_i=\max_{v\in V_i} d(v,G_i)$ とする. $\Delta_{fi}=\max_{v\in V_i} d(v,G_i)/f(v)$ とする. 次式が成立つ.

$$\lceil\Delta/2^i\rceil-1 \le \Delta_i \le \lceil\Delta/2^i\rceil+1$$

$$\lfloor E/2^i\rfloor \le E_i \le \lceil E/2^i\rceil$$

EULER-P(G) の計算時間は $O(|E|)$ である. $c\le\log\Delta_f$であるので 3 行目はアルゴリズム全体で

$$O\left(\sum_{i=0}^{c-1}2^i|E_i|\right) \le O(|E|\log\Delta_f) \le O(|E|\log|E|)$$

時間かかる.

2 行目の PARALLEL-COLOR はアルゴリズム全体で

$$O(2^c\cdot|E_c|\cdot\Delta_{fc}\log|E_c|)$$

$$=O(2^{-c}E\cdot\Delta_f\cdot\log|E|)$$

計算時間かかる.

9 行目の RECOLOR はアルゴリズム全体で

$$O\left(\sum_{i=0}^{c-1}2^i 3|E_i|\cdot(|E_i|+|V|)/\Delta_{fc}\right)$$

$$\le O\left(\sum_{i=0}^{c-1}2^i 3|E_i|\cdot|E|/\Delta_{fc}\right)$$

$$=O(2^c|E|^2/\Delta_f)$$

計算時間かかる.

$2^c=\Delta_f\sqrt{(\log|E|)/|E|}$より本アルゴリズムの計算時間は合計 $O(|E|\sqrt{|E|\log|E|})$ である. (証明終)

6. あとがき

本論文ではグラフ G を高々$u(G)$ 個数の色を用いて f彩色する $O(|E|\Delta_f\log|E|)$ 時間アルゴリズムおよび $O(|E|\sqrt{|E|\log|E|})$ 時間アルゴリズムを与えた. 2 つのアルゴリズムの記憶領域は $O(|V|\Delta_f+|E|)$ である.

今後さらに f彩色アルゴリズムを高速化することが期待される. また高速並列アルゴリズムの研究も望まれる.

グラフの f彩色は更に fg彩色に拡張されている[12]. 本アルゴリズムをさらに拡張し, グラフの fg彩色に応用できることが予想される.

文献

(1) A. V. Aho, J. E. Hopcroft and J. D. Ullman: "The Design and Analysis of Computer Algorithms," Addison-Wesley, Reading, MA. (1974).
(2) E. G. Coffman, Jr, M. R. Garey, D. S. Johnson and A. S. LaPaugh: "Scheduling file transfers," SIAM J. Comput., 14, 3, pp.744-780(1985).
(3) S. Fiorini and R. J. Wilson: "Edge-Colouring of Graphs," Pitman, London (1977).
(4) M. R. Garey and D. S. Johnson: "Computers and Intractability : A Guide to the Theory of NP-Completeness," W. H. Freeman & Co., New York (1979).
(5) H. N. Gabow, T. Nishizeki, O. Kariv, D. Leven and O.Terada: "Algorithms for edge-coloring graphs," Tech.Rep.TRECIS-8501, Tohoku Univ. (1985).
(6) S. L. Hakimi: "Further results on a generalization of edge-coloring", in Graph Theory with Applications to Algorithms and Computer Science, ed. Y. Alavi et al.,pp.371-389, John Wiley & Sons, New York (1985).
(7) S. L. Hakimi and O. Kariv: "On a generalization of edge-coloring in graphs", Journal of Graph Theory, 10, pp.139-154 (1986).
(8) D. S. Hochbaum, T. Nishizeki and D. B. Shmoys: "A better than "best possible" algorithm to edge color multigraphs," Journal of Algorithms, 7, 1, pp.79-104 (1986).

(9) I. J. Holyer: "The NP-completeness of edge colourings," SIAM J. Comput., 10, pp.718-720 (1980).
(10) H. Krawczyk and M. Kubale: "An approximation algorithm for diagnostic test scheduling in multicomputer systems," IEEE Trans. Comput. C-34, 9, pp.869-872 (1985).
(11) S. Nakano, T. Nishizeki and N. Saito "On the f-coloring of multigraphs," IEEE Trans. Circuit and Syst., CAS-35, 3, pp.345-353 (1988).
(12) 中野, 西関, 斎藤: "グラフの fg辺彩色数の上界", 信学論(A), J70-A, 10, pp.1463-1471 (昭62-10).
(13) V. G. Vizing: "On an estimate of the chromatic class of a p-graph," Discret Analiz, 3, pp.25-30 (in Russian) (1964).
(14) V. G. Vizing: "The chromatic class of a multigraph," Kibernetica(Kief), 3, pp.29-39 (1965); Cybernatics, 3, pp.32-41 (1965).