グラフの彩色アルゴリズム

遠藤貴世美　(指導教員：萩田真理子)

1　はじめに

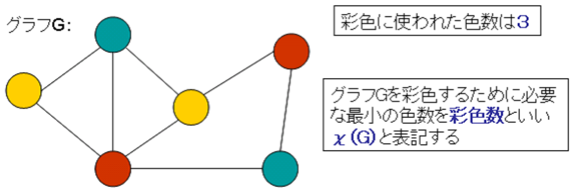
グラフの彩色アルゴリズムとして最も有名な Welsh・Powell のアルゴリズムは、与えられたグラフ G を、G の最大次数 $\Delta(G)+1$ 色以下で彩色する、高速かつ比較的少ない色数で彩色できるアルゴリズムである。これを改良し、G の部分グラフ H の最小次数 $\delta(H)$ の最大値 +1 色あれば必ず彩色でき、計算に必要な時間も計算空間にもそれほど差がない以下のアルゴリズムを提案する。

1．次数の小さい点から順に番号をつけ、$H=G$ とする。2．H の中で次数が最小の点の内一番小さな番号の点 v を選び、$H=H-\{v\}$ とする、この操作をグラフの点の数だけ繰り返す。３．２で順に選び出していった点と逆順に彩色を行う。ただし、使用する色番号はその点の周りに使われていない色番号の中で一番小さい色番号とする。

この二つのアルゴリズムを用いて実際にグラフを彩色した場合に色数にどのような違いが見られるか。また与えられたグラフが平面グラフに限った場合のアルゴリズムなども提案していく。

2　彩色アルゴリズム

グラフの点彩色とは、隣り合う点が互いに異なる色となるようにグラフ全体を塗り分けるというものである。例として以下にグラフ G の彩色を示す。



このようにグラフを彩色する方法としてよく知られている次のアルゴリズムを紹介する。

Welsh・Powell のアルゴリズム

入力：グラフ G ($|V(G)|=n$)

1. $d(v_1)\geq d(v_2)\geq\cdots\geq d(v_n)$ となるように、次数が大きい順に点に番号を付ける。
2. $i=1,2,\cdots,n$ について、以下を繰り返す：
   (a) v_i を G で周りに使われていない最小の色番号でぬる。

出力：グラフ G の点彩色

これは「次数の大きい点から順にできるだけ小さな番号の色で彩色する」アルゴリズムで、この方法で彩色されたグラフの色数は $\Delta(G)+1$ 以下となる。

今回はこのアルゴリズムを改良して得られたものでより少ない色数で彩色できることの多い次のアルゴリズムを提案する。

アルゴリズム 1

入力：グラフ G ($|V(G)|=n$)

1. $d(w_1)\geq d(w_2)\geq\cdots\geq d(w_n)$ となるように、次数が大きい順に点に番号を付け $H=G$ とする。
2. $i=1,2,\cdots,n$ について、以下を繰り返す：
   (a) グラフ H の中で次数が最小の点の内、一番大きな番号の点 v_i を選ぶ。
   (b) $H=H-\{v_i\}$ とする。
3. $i=n,n-1,\cdots,1$ について、以下を繰り返す：
   (a) v_i を H で周りに使われていない最小の色番号でぬる。
   (b) H に v_{i-1} と G での v_{i-1} の周りの辺を加えたグラフを新しく H とする。

出力：グラフ G の点彩色

定理 1
アルゴリズム 1 でグラフ G を彩色したときに使われる色数は、彩色の際に出てくる G の部分グラフ H における $\delta(H)$ の最大値 +1 以下となる。

この定理は，よく知られている以下の定理を含む。

系 1
H を G に含まれる全ての部分グラフとすると、$\chi(G)\leq\max\{\delta(H)\}+1$ となる。

証明
アルゴリズム 1 で彩色に使われた色数を n とする。n 番目の色でぬられた点を v としてアルゴリズム 1 で v まで戻したグラフを H とする。v を彩色するには n 番目の色を使用しなければならないので v は少なくとも次数 $n-1$ のはずであり、v は H で次数が最小の点なので $\delta(H)\geq n-1$ となる。
よって、$\chi(G)\leq n\leq\delta(H)+1\leq\max\{\delta(H)\}+1$ が示された。

3　計算量

計算に必要な時間や計算空間にはどの程度違いが見られるのかを検証していく。

Welsh・Powell のアルゴリズムの計算量

次数が降順になるように点に番号を付ける：計算量 $O(n)$、外部記憶 $O(n)$。周りに使われていない色で順に彩色する：計算量 $O(n^2)$、外部記憶 $O(1)$。

アルゴリズム 1 の計算量

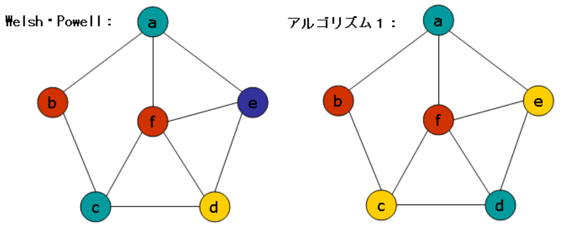
次数が降順になるように点に番号を付ける：計算量 $O(n)$、外部記憶 $O(n)$。最小次数を選び部分グラフを作成：計算量 $O(n^2)$、外部記憶 $O(n^2)$。周りに使われていない色で順に彩色する：計算量 $O(n^2)$、外部記憶 $O(1)$。

アルゴリズム 1 では部分グラフを作成するため計算量・外部記憶共に増加する。しかし彩色の際の計算量がどちらも $O(n^2)$ であることと、もともとグラフ自体の辺情報が $O(n^2)$ であるため、計算量と計算空間はどちらも $O(n^2)$ となる。実際は Welsh・Powell のアルゴリズムと比べてアルゴリズム 1 の方が約 2 倍必要となる。

4　彩色に用いられる色数の違い

実際に Welsh・Powell のアルゴリズムとアルゴリズム 1 でグラフを彩色した場合に用いられる色数にどのくらい差があるのかを見ていく。

例えば下のように頂点番号が与えられていたグラフの場合の彩色は図のようになる。



アルゴリズム 1 は隣接している点を優先的に彩色できるので色数も得をする場合が多い。

ではグラフ全体ではどの程度の差が見られるのか。同形なグラフであっても、上記のように点の番号の付け方で彩色の仕方に違いが出るため $|V(G)| = n$ の場合 $2^{\frac{n(n-1)}{2}}$ 通りのグラフを、それぞれのアルゴリズムを使って彩色した結果を以下に表す。

$\|V(G)\|$	総数	色数が少ないグラフの数	
		Welsh・Powell	アルゴリズム 1
6	2^{15}	8	200
7	2^{21}	2891	29337
8	2^{28}	866813	7213269

アルゴリズム 1 の方が少ない色数で彩色できることが多いとわかる。

5　平面グラフの彩色

アルゴリズム 1 を使い平面グラフを彩色した場合を考える。全ての平面的グラフは次数 5 以下の点を持つことが知られているので、平面グラフ G の任意の部分グラフ H において $\max\{\delta(H)\} + 1 \leq 6$ を満たし、定理 1 より次が成り立つ。

定理 2
平面的グラフをアルゴリズム 1 を用いて彩色すると、必ず 6 色以下で彩色される。

6 色では彩色できることがわかったので、次に平面的グラフを 5 色以下で彩色するアルゴリズムを考える。

補題 1
平面グラフには次数 5 の点 v の周りの点の中に互いに隣接していない 2 点が必ず存在する。

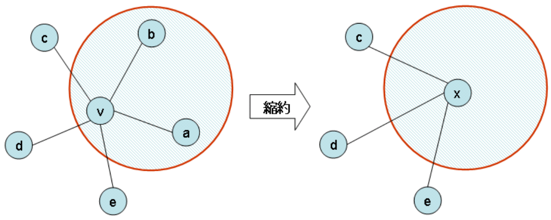
これにより、その互いに隣接していない 2 点を同じ色で彩色すればいいとわかる。

アルゴリズム 2
入力：グラフ G ($|V(G)| = n$)

1. $H = G$ として点がなくなるまで以下を繰り返す：
  (a) グラフ H の中で次数が最小の点 v を選び v の次数によって場合分けを行う。
    i. $d(v) \leq 4$ の場合
      v を取り除いたグラフを新しく H とする。
    ii. $d(v) = 5$ の場合
      v の周りの点のうち互いに隣接していない 2 点 a, b を選び、v, a, b を 1 点 x に縮約したグラフを新しく H とする。
    iii. $d(v) > 5$ 又は　で a, b が見つからない場合
      G は平面的でないと返して終わる。

2. グラフを戻しながら点 v を *(a) (b)* どちらかで彩色する。
  (a) 　の場合は周りに使われていない最小の色番号でぬる。
  (b) 　の場合は縮約された点を戻して x についていた色 C を a, b につけて v を周りにない最小の色番号でぬる。

出力：グラフ G の点彩色



アルゴリズム 2 では次数 5 の点の周りには隣り合わない点が必ず存在するとしている。そのような点が存在しないで全ての点が隣り合っているということは、部分グラフ H が頂点数 5 の完全グラフ K_5 ということになり、以下の Wagner の定理より平面的グラフではないことがわかる。

Wagner の定理
G が平面的グラフであるための必要十分条件は、縮約して K_5 又は $K_{3,3}$ となる部分グラフを含まないことである。

定理 3
平面的グラフをアルゴリズム 2 を用いて彩色すると、必ず 5 色以下で彩色される。

ただし、平面的グラフでなくても $K_{3,3}$ のように 5 色以下で彩色できるものがあるので、アルゴリズム 2 で彩色できたからといって平面的グラフであるとは言えない。

6　まとめと今後の課題

アルゴリズム 1 では、$\delta(H)$ の最大値 +1 色以下で彩色可能。平面的グラフは 6 色以下で彩色可能である。アルゴリズム 2 では、平面的グラフは 5 色以下で彩色可能。アルゴリズム 2 で彩色できないグラフは平面的グラフではない。

今後の課題としては、ただ隣り合う点を異なる色で彩色するだけでなくできるだけ色の偏りが少なくなるように彩色する分散彩色アルゴリズムを作りたい。

参考文献

[1] R.A.Wilson, Graphs, Colourings and the Four-colour Theorem. Oxford University Press Reprinted 2004.