グラフの分散彩色

山口 真実　(指導教員　萩田 真理子)

1　はじめに

並列計算機でシミュレーションをするときに、各計算機で擬似乱数を発生させて用いることがある。

例：核分裂の様子を調べるとき、空間を小さな区域に分割して、それぞれの領域での確率現象を、各計算機で生成した擬似乱数を用いて計算させる。

しかし、近くの現象を扱う計算機が、全く同じ擬似乱数列を生成していたら、それによる偏りがおきてしまう。相関が大きい場所では、なるべく違う関数で生成された擬似乱数を用いたい。何種類かの関数しかないときに、どのように割り振れば良いだろうか？

このような問題を解決するために、グラフの分散彩色アルゴリズムが必要とされている。

2　定義

本稿で扱う言葉の定義を以下のようにする。

定義 1 頂点集合 $V(G)$、辺集合 $E(G)$ のグラフ G について、グラフの各頂点に色番号を割り当てる関数

$$c : V(G) \to C = \{1, 2, \cdots, k\}$$

が、「$xy \in E(G)$ ならば $c(x) \neq c(y)$」 をみたすとき、c を G の彩色または k-彩色という。

定義 2 グラフ G の彩色 $c : V(G) \to C = \{1, 2, \cdots, k\}$ について、

$$cd(c) := \min\{d(x, y) \mid c(x) = c(y), x \neq y\}$$

を彩色距離と呼ぶ。

定義 3 彩色距離が $cd(c) > d$ をみたす彩色が存在するために必要な色数の最小値

$$\chi_d(G) := \min\{k | G \text{ に } cd(c) > d \text{ を満たす } k\text{-彩色 } c \text{ が存在 }\}$$

を G の d-分散彩色数とよび、χ_d であらわす。

命題 1 グラフ G の d-分散彩色数は、G の距離 d 以下の頂点間を辺で結んだグラフ G^d の彩色数と等しい。つまり、グラフ G^d を

$$\begin{aligned} V(G^d) &= V(G) \\ E(G^d) &= \{xy | x, y \in V(G), d_G(x, y) \leq d\} \end{aligned}$$

で定義されるグラフとすると、

$$\chi_d(G) = \chi(G^d)$$

をみたす。

3　評価方法

分散彩色を評価する指標として、次のように定義される彩色の重みを提案する。

定義 4

$$w(c) := \sum_{c(x)=c(y)} \frac{1}{d(x, y)} \tag{1}$$

$$z(c) := \sum_{c(x)=c(y)} \frac{1}{d(x, y)^2} \tag{2}$$

$w(c)$ は同色頂点間距離の逆数の和、$z(c)$ は同色頂点間距離の逆数の 2 乗の和である。

同色頂点間距離が大きくなるにつれ、式 1、2 の値は小さくなる。つまり、彩色結果を評価したときにこれらの値が十分に小さければ、同色頂点間距離が大きい、良い分散彩色を与えていることになる。

4　彩色アルゴリズム

アルゴリズム 1 （WelshPowell のアルゴリズム）

このアルゴリズムは、高速に $\chi(G)$ に近い値で彩色をするアルゴリズムである。

次数の高い頂点から順に、そこから距離 1 の頂点に使われていない最小の色番号で彩色する。

アルゴリズム 2 （同色頂点間距離が必ず d 以上になるアルゴリズム）

このアルゴリズムは彩色したグラフに対し、$d-$ 分散彩色を与える。また、アルゴリズム中で *WelshPowell* のアルゴリズムをしており、$\chi_d(G)$ を求めることができる。

入力：グラフ G ($|V(G)| = n$),
　　彩色に用いる色の数 k

1. 入力されたグラフに対し、各頂点間の距離を求め、参照しやすいように値を保存する。（以下、*list*1 と呼ぶ。）
2. 頂点間距離が $d(= 1)$ ならば、辺を追加し、このグラフを G^d とする。（辺の追加時には、*list*1 を参照する。）
3. *WelshPowell* のアルゴリズムでグラフ G^d を彩色する。
4. 彩色数が k より少ないならば、d の値を 1 増やし、2, 3 を繰り返す。
5. 彩色数が k を超えたら、1 つ前の状態を返して終了。

出力：グラフ G の点彩色結果

アルゴリズム 3 （今回作成したアルゴリズム）

入力：グラフ G ($|V(G)| = n$),
　　彩色に用いる色の数 k

1. 各頂点間の距離を求め、参照しやすいように値を保存する。（以下、*list*1 と呼ぶ。）
2. すべての頂点の色を c_1 とする。
3. *list*1 を参照しながら、同色頂点間距離の逆数の和の *list* を作成する。（以下、*list*2 と呼ぶ。）
4. 頂点の色を変更することで、彩色の重み $w(c)$ が小さくなるならば、色を変更する。
5. 色の変更にともない、*list*2 の更新を行う。
6. 4、5 を n 回繰り返したら終了。

出力：グラフ G の点彩色結果

アルゴリズム 4 （今回作成したアルゴリズム）
アルゴリズム 3 の彩色の重み $w(c)$ を $z(c)$ にしたものをアルゴリズム 4 とする。

5 計算量

各アルゴリズムの計算量について考える。

WelshPowell 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 $O(n)$。周りに使われていない色で順に彩色する：計算量 $O(n^2)$。

同色頂点間距離が必ず d 以上になるアルゴリズム (アルゴリズム 2) の計算量
list1 の作成:$O(n^3)$。グラフ G^d の作成:$O(n^2)$。次の 2 つは WelshPowell のアルゴリズムと同じである。次数が降順になるように点に番号を付ける：計算量 $O(n)$。周りに使われていない色で順に彩色する：計算量 $O(n^2)$。グラフ G^d と WelshPolwell のアルゴリズムを指定した色数を超える手前まで $G^d(d = 2, 3, \cdots)$ で繰り返す。

アルゴリズム 3 の計算量
list1 の作成:$O(n^3)$。頂点の色の初期化:$O(n)$。list2 の作成:$O(n^2)$。色の変更:$O(n)$。色の変更に伴う list2 の更新:$O(n)$。色の変更と、それに伴う list の更新については、n 回繰り返すので、この部分の計算量は $O(n^2)$ となる。

アルゴリズム 4 の計算量
彩色の重みを $w(c)$ から $z(c)$ に変更しただけなので、アルゴリズム 3 と同様。

6 評価 1

アルゴリズム 2 は彩色距離を大きくすることができ、アルゴリズム 3 は同色頂点間距離を大きくすることができる。

この 2 つのアルゴリズムの彩色結果に対し、彩色の重み $w(c)$ を求めることで同じ色数で彩色したときの違いを確認した。

6.1 方法

1. 頂点数が 100、辺の本数が最大 495 本のランダムな連結グラフを作成。
2. 作成したグラフをアルゴリズム 2、3 で彩色。
3. それぞれの結果に対し、彩色の重み $w(c)$ を求める。

ここでは、同じ色数で塗った場合の結果を比較したかったので、アルゴリズム 2 の d の値を $d = 2$ と固定にした。

6.2 結果

実行結果を表 1 に示す。実装上の都合により、表 1 に書かれている彩色の重みは、定義 4 で定義した彩色の重み $w(c)$ の 2 倍の値になっている。

表 1 より、同じ色数で彩色してもアルゴリズム 3 の彩色結果のほうが彩色の重みが小さくなっていることが確認できる。

これは、アルゴリズム 2 が WelshPowell を用いて彩色しているため、同じ色を沢山使う可能性があるのに対し、アルゴリズム 3 は同色頂点間を大きくするために、できるだけばらばらな色を使おうとしているためである。

表 1：アルゴリズム 2、3 の結果に対する彩色の重み

辺の数	彩色数	アルゴリズム 2	アルゴリズム 3
475	28	91.999	91.499
471	29	91.333	87.333
486	28	94.666	90.333
476	28	91.999	90.499
469	29	93.666	85.499

7 評価 2

今回、彩色の重みとして $w(c)$、$z(c)$ の 2 つを提案している。これらはどちらのほうがより良い結果につながる評価指標となるのだろうか？

7.1 方法

1. グラフを作成する。
2. WelshPowell のアルゴリズムで彩色し、彩色数を求める。ここで求めた彩色数 +1 をアルゴリズム 3、4 で用いる。
3. アルゴリズム 3、4 で彩色する。
4. 彩色結果に対し、彩色の重み $w(c)$、$z(c)$ を求める。

これを $|V(G)| = n$ の場合 $2^{\frac{n(n-1)}{2}}$ 通りのグラフに対して行った。これは、同形なグラフであってもアルゴリズム 3、4 ともに、頂点番号が小さいものを優先して塗るようになっているため、点の番号のつけ方で彩色の仕方に違いが出ることが予想されるからである。

7.2 結果

$n = 3, 4, 5, 6, 7$ の場合、求めた彩色の重み ($w(c)$、$z(c)$) に違いは見られなかった。どちらのアルゴリズムでも、彩色結果が同じになったためである。

8 まとめと今後の課題

これまでに、

- 分散彩色アルゴリズムの提案
- 分散彩色の評価方法として、彩色の重み $w(c)$、$z(c)$ を用いることの提案
- アルゴリズム 3 の評価

を行った。今後は、

- アルゴリズム 4 の評価
- より頂点数の大きなグラフに対して評価 2 の実行
- アルゴリズム 3、4 の比較

などを行いたい。

参考文献

[1] Martin Aigner : Discrete Mathematics