Parameterized Complexity of Flood-Filling Games on Trees

木上における Flood-Filling ゲームのパラメータ複雑度

U. dos Santos Souza[1], F. Protti[1] and M.D. da Silva[2]

[1]Institute of Computing
[2]PURO/ICT
Fluminense Federal University, Niterói, RJ,Brazil
COCOON 2013, LNCS 7936, pp. 531–542, 2013.

発表者：周・伊藤研究室　学部 4 年　菅崎 拓真

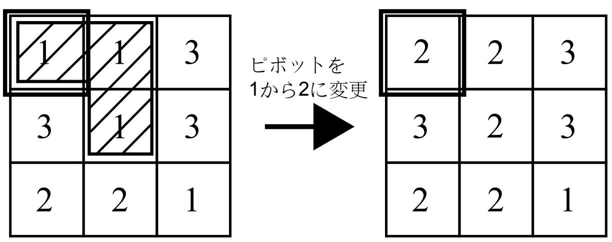
1　はじめに

Flood-Filling ゲームとは，ある色分けされた区画を決められたルールに基づいて一色に塗りつぶす一人用のゲームの総称であり，塗りつぶしのルールによって Flood-It，Free-Flood-It，Multi-Flood-It などのゲームが存在する．

Flood-It とは，$n \times m$ の色分けされたグリッド上で行うゲームである．Flood-It はグリットの一番左上にあるマスを基準 (ピボット) として進めて行くゲームであり，1 回の操作でピボットをある色から別のある色へ変更する．このときピボットから縦横にピボットと同じ色のマスのみを辿って到達できる全てのマスの色が同時に変わる．

図 1 に操作の例を示す．操作前図 1(a) の射線で表した 3 マスがこのときの操作で同時に色が変わるマスであり，ここにピボットの色を 1 から 2 へ変える操作を適用すると，操作後図 1(b) のグリッドが得られる．この操作を繰り返してグリッドを一色に塗りつぶすことが目的である．このとき，操作数が最小となるような操作列を求めることが Flood-It の問題である．

Free-Flood-It とは大枠は Flood-It と同様だが，1 回の操作ごとにグリッド上の全てのマスの中から自由にピボットを選定してよい．とするルールである．Multi-Flood-It も大枠は Flood-It と同様だが，あらかじめ定めた複数の固定されたピボット群を用いるルールである．



(a) (b)

図 1: 1 回の操作の例．マス目の中の数字は色を表す．

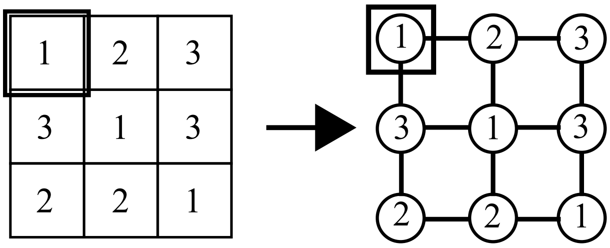


図 2: グリッドとグラフの対応．数字は色を表す．

本論文内では既知の結果と合わせて，これら 3 つの塗りつぶしに対して様々な知見を与えている．

Flood-It の問題に関して，既知の結果としてグリッドの大きさが $n \times n$ かつ 3 色以上ならば NP 困難 [1]，$3 \times n$ かつ 4 色以上ならば NP 困難，$2 \times n$ ならどんな色数だとしても多項式時間で解けることが知られている [3].

また，Flood-It はグリッド上だけではなく，色分けされた頂点とそれらを繋ぐ辺の上でも考えることができる (図 2)．これはグラフへの一般化であり，様々な種類のグラフにおいて，Flood-It を考えることができる．既知の結果としてサイクル上では多項式時間で解ける一方，木上においては NP 困難であることが知られている [2].

本論文では主に木上の Flood-It について扱い，入力グラフをピボットからの距離 d が高々2 であるような木に制限したとしても依然として NP 困難であること．ピボットからの距離 d が定数であるような木において，色の数 c に対する FPT であることを実際に FPT アルゴリズムを与えることで示した．また，Flood-It からの帰着を用いて Free-Flood-It，Multi-Flood-It の困難性や RSCS という文字列の分野との関係も示した．

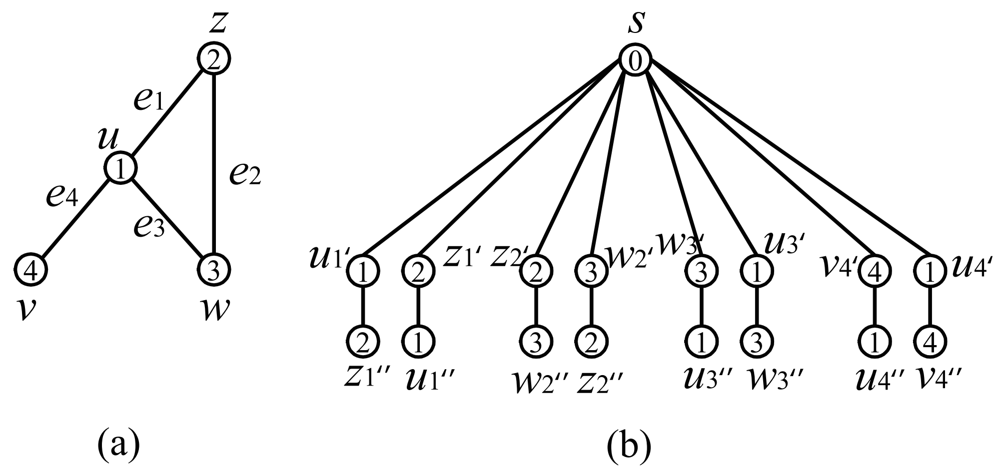


図 3: (a) グラフ G，(b)G から得られる関連木 T． 頂点内の数字は色を表す．

2 定義

解くべき Flood-It の問題について，以下のように定義する．

- 入力：頂点が色分けされたグラフ $G = (V, E)$, ピボットとなる頂点 $p \in V$
- 出力：グラフ上の頂点を一色に塗りつぶすための最小の操作列．

以下ではグラフとして入力された木をピボット p を根とする根つき木として扱う．また，入力された木の頂点数を n, 木の高さを d, 使用されている色の種類数を c とする．

3 本論文の結果

3.1 木上の Flood-It

この節では木上の Flood-It について以下の定理を証明する．

定理 1 Flood-It はピボットから高々距離 $d = 2$ のところまでにしか葉をもたない木上では NP 困難である．

証明 頂点被覆問題からの帰着を用いる．グラフ G 内のサイズ k の頂点被覆が存在する時，かつその時に限り，関連木 T 上の $n+k$ 回の操作による Flood-It が存在する．

グラフ G を $G = (V, E)$,$|V| = n$,$|E| = m$ とすると関連木 T は以下のように作られる．

- ピボット根 s, 色 c_s を作る
- グラフ G の全ての辺 $e_i = uv$ について，辺を構成する頂点集合 $E_i = \{u_i', v_i', u_i'', v_i''\}$ のベキ集合を T に追加する．u_i',v_i' は s の子であり，v_i'' は u_i' の子，u_i'' は v_i' の子である．
- グラフ G 上の全ての点に異なる色 c_u を定義し，関連木 T における u_i',u_i'' の色はそれぞれの c_u と同じとする (色は引き継ぐ)．

図 3 にグラフ G と関連木 T の例を示す．

G はサイズ k の頂点被覆 V' をもつと仮定する．関係木の作成法より塗りつぶされていない頂点の集合は n 色である．n 色全てを用いた n 回の操作を行うとそれぞれの部分木 E_i は必ず s と同じ色になっていない頂点を一つだけ含む．

ここで n 回の操作を以下のように行う．始めに V' の頂点に割り当てられている色で操作を行い，その後 残りの操作を行う．G のすべての辺は少なくとも一方は V' の端点であるため，これら n 回の操作により s と同じ色でない T の頂点は V' に関連付けられた色を持つ．$|V'| = k$ より残り k 回の操作が必要である．

ゆえに Flood-It には $n + k$ 回の操作が必要である，と示された．

次に T が $n + k$ 回の操作で塗りつぶすことができると仮定する．はじめに T が c_s と色集合 C からなる n 色からのみの色を含むとする．故に C のそれぞれの色は少なくとも一回は Flood-It の色に選定される．

C 内の色を 2 グループに分ける．

1. 2 回以上塗りつぶされる色のグループ
2. 1 回だけしか塗りつぶされない色のグループ

部分木 E_i を塗りつぶすために最初と最後の操作は第一グループの色で塗りつぶされる．故に一般性を失わずに以下の手順で $n + k$ 回の操作を規定できる．

a). 一番始めの操作を第一グループの色から選定する．

b). 第二グループの色の操作を行う．

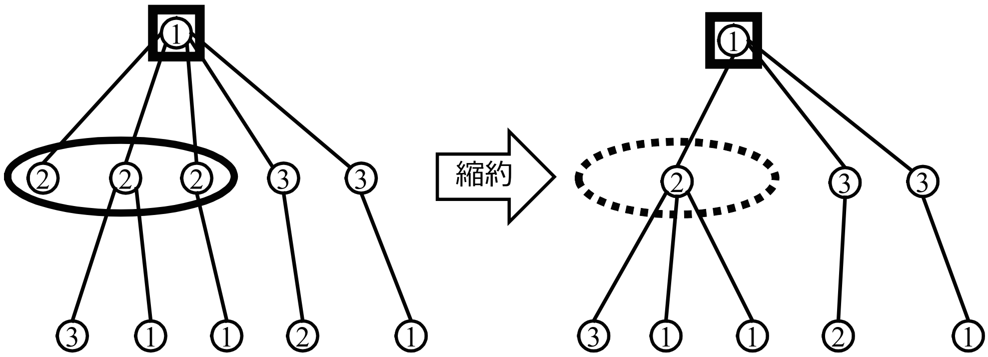


図 4: 縮約の様子．この操作を子をもつ頂点全てに適用する．

c). 残りの操作を第一グループの色から選定する．

n 回の操作は (a) と (b) に対応し（これらは各色ごとの n 回の操作である），部分木 E_i のひとつの頂点は塗りつぶされていないままである．加えて，s と同色でない頂点は k 色である．

T のそれぞれの色は G の異なる頂点を表し，かつ 作成法より E_i は G の異なる辺と関係するため，これら k 色はサイズ k の G の頂点集合 (サブセット) に対応する．すなわち G の頂点被覆である．

頂点被覆問題は NP 困難であることが知られているので定理 1 は示された．□

3.2 色の数 c に対する FPT

定理 1 より木の高さ $d = 2$ のような簡単な木上であっても依然として NP 困難であると示された．NP 困難なので最適解を高速に求めることは難しい，と分かった．ではここでアプローチを変えて，最適解をやや高速に求めることを考える．

このアプローチに則した手法を **FPT(Fixed Parameter Tractability) アルゴリズム**と呼ぶ．FPT アルゴリズムとはパラメータが小さければ最適解がやや高速に求まるアルゴリズムである．具体的にはパラメータ c に対して計算時間が $f(c) \cdot n^{O(1)}$ で走るアルゴリズムのことであり，今回は色の数 c をパラメータとして，計算時間 $O(c^{c^d} + n)$ でグラフを塗りつぶすための最小の操作が求まるアルゴリズムを導いた．以下定理 2 とともに説明していく．

定理 2 ピボットから高々定数 d の距離までに葉をもつ木上の Flood-It は色数 c が小さければ多項式時間である (FPT)．

証明 実際に以下のように FPT アルゴリズムを与える．

T を n 個の頂点とピボット p からなる木とする．

1. $T' := T$ とする
2. T' のピボット p とその子のうち，色がピボット p と同じものをひとつの頂点にまとめる．
3. T' のピボット p の子のうち，色が同じもの同士をひとつの頂点にまとめる．
4. T' のピボット p の子のうち葉でない限り手順 2 と手順 3 を再帰する．

手順 2 と手順 3 を**縮約**といい，この縮約の手法により元の T から，より頂点数が少ない縮約木 T' を導くことがこのアルゴリズムのアイデアである．縮約の手法は元のグラフ T において常に同じ操作で塗りつぶすことができるため適用できるものである．よって縮約の手法を行っても最終的な塗りつぶしの操作列には影響せず，縮約木 T' での最小の操作列はそのまま元の T の最小の操作列として用いることができる．手順 3 の縮約の様子の例を図 4 に示す．

このアルゴリズム適用後，T' のすべての点は高々色数 $c - 1$ 個だけの子を持つ．故に T' は $O(c^d)$ 個の頂点をもち，かつそれらは多項式時間で塗りつぶすことができる．

ここで 1 回の操作で少なくとも 1 つの点が新たに塗りつぶされる操作のみを考えると，最悪でも T' の頂点数 $O(c^d)$ 回だけ操作を行えば T' を塗りつぶすことができる．各回それぞれの操作について考えられる色は $c - 1$ 通りある．よって全体で考えられる操作列は $O(c^{c^d})$ 通りある．

最小の操作数のものはこの中に存在する．総当りで $O(c^{c^d})$ 通り全て試すことで，グラフを一色にするもののうち最小のものを求めることができる．

T' を求めるために元の T の頂点数 n に対して $O(n)$ 分だけ計算時間がかかるので，全体としてこのアルゴリズムは $O(c^{c^d} + n)$ で走る．これは多項式時間であり，よってこのアルゴリズムは FPT アルゴリズムである．□

4 まとめ

本論文では既知の結果に加えて，頂点被覆問題からの帰着を用いてピボットからの距離 d が高々2 であるような木でさえも依然として NP 困難であること．ピボットからの距離 d が定数であるような木において，色の数 c に対する FPT であることを実際に FPT アルゴリズムを与えることで示すことができた．

参考文献

[1] Arthur, D., Clifford, R., Jalsenius, M., Montanaro, A., Sach, B.: The Complexity of Flood Filling Games. In: Boldi, P., Gargano, L. (eds.) FUN 2010. LNCS, vol. 6099, pp. 307–318. Springer, Heidelberg (2010)

[2] Lagoutte, A., Noual, M., Thierry, E.: Flooding Games on Graphs, HAL: hal–00653714 (December 2011)

[3] Meeks, K., Scott, A.: The Complexity of Flood-Filling Games on Graphs. Discrete Applied Mathematics, vol. 160, pp. 959-969 (2012)