Theoretical Computer Science 576 (2015) 102–116

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Tractability and hardness of flood-filling games on trees

Michael R. Fellows [a], Uéverton dos Santos Souza [b,c,*], Fábio Protti [b], Maise Dantas da Silva [d]

[a] Charles Darwin University, Darwin, Australia
[b] Institute of Computing – Fluminense Federal University, Niterói, Brazil
[c] Federal Center of Technological Education of Rio de Janeiro, Rio de Janeiro, Brazil
[d] Institute of Computing Science – Fluminense Federal University, Rio das Ostras, Brazil

ARTICLE INFO

Article history:
Received 1 September 2014
Received in revised form 6 February 2015
Accepted 8 February 2015
Available online 17 February 2015
Communicated by D. Peleg

Keywords:
Flood-It
Free-Flood-It
Shortest common supersequence
NP-hardness
W[1]-hardness
Fixed-parameter tractability

ABSTRACT

This work presents new results on flood-filling games, Flood-It and Free-Flood-It, in which the player aims to make the board monochromatic with a minimum number of flooding moves. A flooding move consists of changing the color of the monochromatic component containing a vertex p (the pivot of the move). These games are originally played on grids; however, when played on trees, we have interesting applications and significant effects on problem complexity. In this paper, a complete mapping of the complexity of flood-filling games on trees is made, charting the consequences of single and aggregate parameterizations by: number of colors, number of moves, maximum distance from the pivot, number of occurrences of a color, number of leaves, and difference between number of moves and number of colors.

We show that Flood-It on trees and Restricted Shortest Common Supersequence (RSCS) are analogous problems, in the sense that they can be translated from one to another, preserving complexity issues; this implies interesting FPT and W[1]-hard cases to Flood-It. Restricting attention to phylogenetic colored trees (where each color occurs at most once in any root-leaf path, in order to model phylogenetic sequences), we also show some impressive NP-hard and FPT results for both games. In addition, we prove that Flood-It and Free-Flood-It remain NP-hard when played on 3-colored trees, which closes an open question posed by Fleischer and Woeginger. Finally, we present a general framework for reducibility from Flood-It to Free-Flood-It; some NP-hard cases for Free-Flood-It can be derived using this approach.

1. Introduction

Let G be a graph equipped with a vertex-coloring ω. A *monochromatic path* of G is a path in which all the vertices have the same color, and two vertices $a, b \in V(G)$ are said *m-connected* if there is a monochromatic path between them, i.e., a and b are in the same monochromatic component (maximal monochromatic connected subgraph). A flooding move $m = (p, c)$ in G consists of changing to c the color of a pivot vertex $p \in V(G)$, and of all vertices m-connected to p. The problem of determining the minimum number of flooding moves to make the graph monochromatic is called *Free-Flood-It*.

* Corresponding author at: Federal Center of Technological Education of Rio de Janeiro, Rio de Janeiro, Brazil.
E-mail addresses: michael.fellows@cdu.edu.au (M.R. Fellows), usouza@ic.uff.br (U.S. Souza), fabio@ic.uff.br (F. Protti), maisedantas@id.uff.br (M. Dantas da Silva).

http://dx.doi.org/10.1016/j.tcs.2015.02.008
0304-3975/

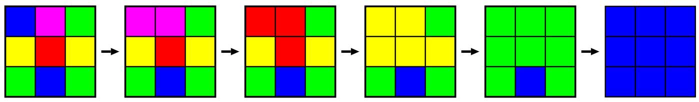

Fig. 1. An optimal sequence of moves to flood a 3×3 grid when playing Flood-It.

Additionally, the problem of determining the minimum number of flooding moves to make the graph monochromatic using a fixed vertex as the pivot for all moves is called *Fixed-Flood-It*, or just *Flood-It*.

The computational problems Flood-It and Free-Flood-It are generalizations of two combinatorial games named alike, where the input underlying graph is an $m \times n$ grid. In Flood-It on grids, the top left vertex is the fixed pivot of the game. Fig. 1 shows a sequence of moves to flood a 3×3 grid colored with five colors.

Many complexity issues on Flood-It and Free-Flood-It have recently been investigated. In [5], Arthur, Clifford, Jalsenius, Montanaro, and Sach show that Flood-It and Free-Flood-It are NP-hard on $n \times n$ grids colored with at least three colors. Meeks and Scott [22] prove that Free-Flood-It is solvable in polynomial time on $1 \times n$ grids, and also that Flood-It and Free-Flood-It remain NP-hard on $3 \times n$ grids colored with at least four colors. To the best of the authors' knowledge, the complexity of (Free-)Flood-It on $3 \times n$ grids colored with three colors remains an open question. Free-Flood-It is known to be solvable in polynomial time on 2-colored graphs [5,19,22]. In [22], Meeks and Scott show that Free-Flood-It on paths can be solved in $O(n^6)$ time. Free-Flood-It on cycles can also be solved in polynomial time [12,24]. Clifford, Jalsenius, Montanaro, and Sach present in [5] a polynomial-time algorithm for Flood-It on $2 \times n$ grids. In [23], Meeks and Scott show that Free-Flood-It remains NP-hard on $2 \times n$ grids and it is fixed-parameter tractable when parameterized by the number of colors. Fleischer and Woeginger [10] proved that Flood-It is NP-hard on trees. In [33], Souza, Protti and Dantas da Silva describe polynomial-time algorithms to play Flood-It on C_n^2 or P_n^2 (the second power of a cycle or a path on n vertices) and $2 \times n$ circular grids, and also show that Free-Flood-It is NP-hard on C_n^2, P_n^2 and $2 \times n$ circular grids.

There is a nice connection between the complexity of Free-Flood-It on trees and the complexity of the problem on general graphs. Meeks and Scott have shown that the minimum number of moves required to flood any given graph G is equal to the minimum, taken over all spanning trees T of G, of the number of moves required to flood T [24].

Applications of flood-filling games. Since the 90's, an increasing number of papers have dealt with biological applications as combinatorial problems. Vertex-colored graph problems have several applications in bioinformatics [7]. The problem of deciding whether a c-vertex-colored graph G is a subgraph of an interval graph properly colored by c (Colored Interval Sandwich Problem, or Intervalizing Colored Graphs) [3] has applications in DNA physical mapping [9,15] and in perfect phylogeny [21]; vertex-recoloring problems appear in protein-protein interaction networks and phylogenetic analysis [4, 27]; the Graph Motif Problem [7], which aims to determine whether a graph G has a connected subset of vertices with a bijection between its colors and set of colors M, was introduced in the context of metabolic network analysis [18]; and the Triangulating Colored Graph Problem [3], which examines vertex-colored graphs that can be triangulated without the introduction of edges between vertices of the same color, it is polynomially equivalent to the Perfect Phylogeny Problem [16].

Flood-Filling games on colored graphs are also related to many problems in bioinformatics. As shown in this work, Flood-It played on trees is analogous to a restricted case of the Shortest Common Supersequence Problem [17]. Consequently, these games inherit from the Shortest Common Supersequence Problem many applications in bioinformatics, such as: microarray production [29], DNA sequence assembly [2], and a close relationship to multiple sequence alignment [31]. Some problems and properties on strings and sequences are more easily observed when translated to a flood-filling dynamics, as done in [7], where a variant of the SCS problem is proved to be fixed-parameter tractable via an argument based on a translation of the problem in terms of the Flood-It game. Besides that, some disease spreading models, described in [1], work in a similar way to flood-filling games.

Flood-It played on trees can also be applied to scheduling. Each color corresponds to an operation in the sequential process of manufacturing an object. In the input tree T, paths from the pivot to the leaves correspond to the manufacturing sequences for a number of different objects that share the same production line. A flooding of T then corresponds to a schedule of operations for the production line that allows all of the different objects to be manufactured. A survey on scheduling of manufacturing operations is the paper [26].

Additional definitions and notation. A subgraph H of G is *adjacent* to a vertex $v \in V(G)$ if v has a neighbor in $V(H)$. A *flood move*, or just *move*, is a pair $m = (p, c_i)$ where p is the *pivot* of m (the vertex chosen to have its color changed by m), and c_i is the new color assigned to p; in this case, we also say that color c_i is *played in move m*. In Flood-It all moves have the same pivot. A subgraph H is said to be *flooded* when H becomes monochromatic. A (free-)flooding is a sequence of moves in (Free-)Flood-It which floods G (the entire board). An optimal (free-)flooding is a flooding with minimum number of moves. A vertex v is *flooded by a move m* if the color of v is played in m and v becomes m-connected to new vertices after playing m. In Free-Flood-It, a move $m = (p, c_i)$ *is played on subgraph H* if $p \in V(H)$. We denote by $\Pi \propto^f \Pi'$ a reduction from a problem Π to a problem Π' via a computable function f. We present below the formal definitions of the two flood-filling games studied in this work.