Flooding games on graphs

Aurélie Lagoutte, Mathilde Noual, Eric Thierry

Flooding games on graphs

A. Lagoutte[a], M. Noual[b], E. Thierry[b,*]

[a] *ENS Cachan, Computer Science Department, France*
[b] *LIP, Université de Lyon, CNRS UMR5668, ENS Lyon, UCB Lyon 1, France*

Abstract

We study a discrete diffusion process introduced in some combinatorial puzzles called Flood-It, Mad Virus or HoneyBee that can be played online [1, 2, 4] and whose computational complexities have recently been studied [5, 7, 12, 8]. Originally defined on regular boards, we show that studying their dynamics directly on general graphs is valuable: we synthesize and extend previous results, we show how to solve Flood-It on cycles by computing a poset height and how to solve the 2-Free-Flood-It variant by computing a graph radius.

Keywords: combinatorial puzzle, colored graph, neighborhood contraction

1. Introduction

There exists a family of combinatorial games (Flood-It [1], Mad Virus [2], Honey-Bee [4]) which rely on a common discrete diffusion dynamics. We study here the graph versions of those "flooding games" that have been originally defined on boards whose tiles are regular polygons (squares for Flood-It, hexagons for Mad Virus and Honey-Bee).

The game called Flood-It is played by one player on a k-colored board where one particular tile s (called *flooding tile*) is fixed. A *colored component* is a maximal set of connected tiles with the same color. The aim of the game is to "flood" the entire board as fast as possible. At each move, the player chooses a color c. The colored component containing s is changed into color c and as a result merges with all adjacent c-colored components. After a finite number of moves, all tiles end with the same color, the board is "flooded" (*i.e.* monochromatic). A sequence of moves is here a sequence of colors. Free-Flood-It [3] is a version of Flood-It in which the player may also choose the flooding tile at each move. A sequence of moves is thus an ordered list of couples (c_i, t_i) such that c_i is the color and t_i the tile that are chosen at move number i. In both versions, a *flooding sequence* is a sequence of moves flooding the entire board (see Figure 1).

Those flooding games have a natural generalization to *connected undirected graphs*. In a k-colored connected graph $G = (V, E, \kappa)$ with coloration $\kappa : V \to \{1, \ldots, k\}$, vertices of V (resp. edges of E) play the same role as tiles (resp. edges of tiles) of the board. Thus, a c-colored component C of G ($1 \leq c \leq k$) is a maximal set of connected vertices

*LIP, ENS Lyon, 15 parvis René Descartes, 69342 Lyon Cedex 07, France.
Email address: eric.thierry@ens-lyon.fr (E. Thierry)

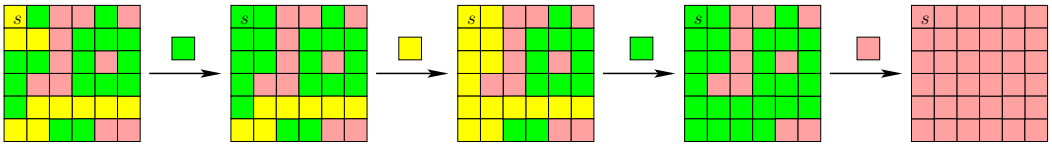

Figure 1: Flood-It on a square board: a flooding sequence green,yellow,green,pink.

that have the same color c, and a move consists in choosing a flooding vertex and a new color for its colored component. The problem `Flood-It` (resp. `Free-Flood-It`), denoted `FI` (resp. `FFI`) takes a k-colored connected graph as input and aims at finding a shortest flooding sequence. When the parameter k is fixed, the problem is called k-`Flood-It` (resp. k-`Free-Flood-It`), denoted $k-$`FI` (resp. $k-$`FFI`).

Board flooding games can be seen as particular instances of graph flooding games by considering the graphs of adjacent tiles as illustrated in Figure 2.

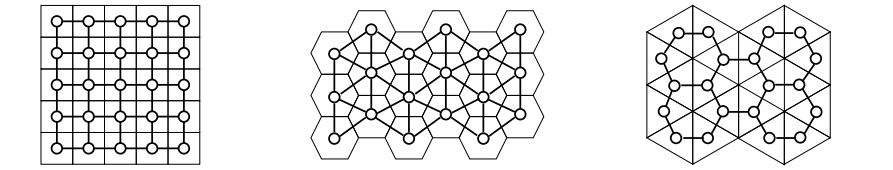

Figure 2: Three types of boards and their underlying graphs (in bold) where each vertex corresponds to a tile.

Recently several studies have investigated the computational complexity for solving those puzzles. They first focused on $n \times n$ square boards [3], and then quickly tackled the problem on arbitrary graphs. Table 1 sums up the cases whose complexities have been classified either as polynomial or NP-hard. Some results have been proved independently by several authors. The references also contain approximability results for the shortest flooding sequences [3, 5, 12], as well as bounds on its length [3, 5].

In this paper, we present new results on those flooding games on graphs, including unpublished results of [11] (in bold in Table 1).

In Section 2, we put forward hard instances of k-`FI` and k-`FFI` based on trees and explain how they encapsulate and extend previously known results. All those hard instances have at least three colors and a vertex of degree at least three. It is then shown that as soon as the number of colors or the maximum degree of the graph is bounded by 2, one can solve the puzzle in polynomial time thanks to shortest or longest path computations. In particular we settle the complexity of 2-`FFI` that was left open in [3] for square boards (which has been solved independently and almost simultaneously in [5, 11, 12]). Section 3 concludes the paper with some open questions.

2. Complexity results

For any k-colored graph $G = (V, E, \kappa)$ we define the *reduced graph* $\widetilde{G} = (\widetilde{V}, \tilde{E}, \tilde{\kappa})$ in which all c-colored components ($c \in \{1, \ldots, k\}$) have been reduced to one c-colored

Classes of graphs	FI	k-FI	FFI	k-FFI
general graphs		NP-hard if $k\geq 3$ (see below) P if $k=2$ (trivial)		NP-hard if $k\geq 3$ (see below) **P if** $k=2$ [11, 12]
$h \times n$ square grids	NP-hard if $h\geq 3$ [5] P if $h=2$ [5]	NP-hard if $h=n$ and $k\geq 3$ [3] NP-hard if $h\geq 3$ and $k\geq 4$ [12]	NP-hard if $h\geq 3$ [5] NP-hard if $h=2$ [13]	NP-hard if $h\geq n$ and $k\geq 3$ [3] NP-hard if $h\geq 3$ and $k\geq 4$ [12] P if $h=n$ and $k=2$ [5] P if $h=2$ [13]
paths ($1 \times n$ grids)	**P** [7, 11]		P [8, 12]	
cycles	**P** [11]		P [8]	
trees		**NP-hard if** $k\geq 3$ [7, 11]		**NP-hard if** $k\geq 3$ [8, 11]
co-comparability graphs	P [7]			
split graphs	NP-hard [7]	P [7]		
triangular grids		NP-hard if $k\geq 6$ [14] **NP-hard if** $k\geq 3$ [11]		**NP-hard if** $k\geq 3$ [11]
hexagonal grids		**NP-hard if** $k\geq 3$ [11]		**NP-hard if** $k\geq 3$ [11]

Table 1: P vs NP-hard classification of Flood-It and its variants, for some classes of graphs

vertex and any two vertices $u, v \in \widetilde{V}$ are connected if and only if they correspond to adjacent colored components of G. By the maximality of the colored components of G, the extremities of any edge in $\widetilde{G}$ have different colors, *i.e.* this coloring $\widetilde{\kappa}$ defines a *proper* k-coloration of $\widetilde{G}$. Note that the reduced graph $\widetilde{G}$ is a minor of G since it can be obtained by edge-contractions. It can be easily constructed from G in $\mathcal{O}(|E|)$ time: any graph traversal which identifies the colored components by first exploring vertices with an identical color is suitable (*e.g.* see [11]). In addition, a sequence $\mathcal{S}$ of moves obviously floods the entire graph G if and only if $\mathcal{S}$ also floods entirely its reduced version. In terms of reduced graphs, a flooding sequence is a sequence which ends on a reduced graph which is a singleton. A flooding sequence for Free-Flood-It and 2 colors is illustrated in Figure 3 with both the original dynamics and the associated sequence of reduced graphs. Note that for 2 colors, each move is fully specified by the flooded vertex whose new color is forced (since all its neighbors have the same color).

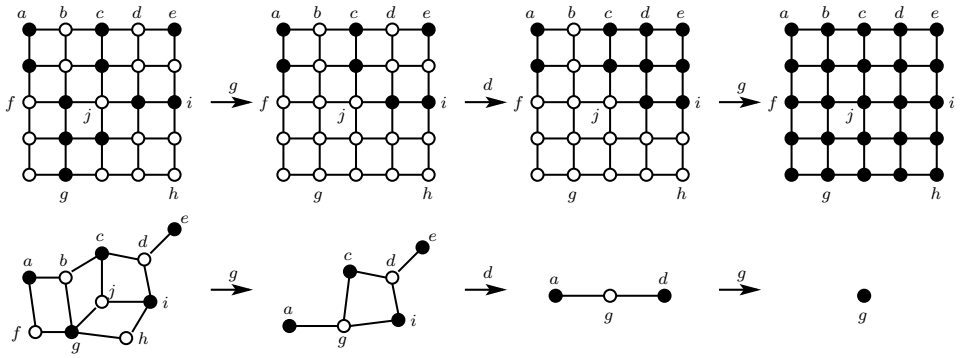

Figure 3: A 2-FFI instance and a sequence of moves over an initial coloring along with the sequence of associated reduced graphs.

First we present hardness complexity results and relate them to some recently published results by revisiting their proofs and extending them to get new results. We call *willow trees* the trees composed of a set of disjoint paths joined by a vertex adjacent to one extremity of each path.

Proposition 1. *For any fixed $k \geq 3$, the decision versions of k-*`FI` *and k-*`FFI` *are NP-complete on willow trees.*

Proof. The idea is to use a reduction from a variant of the problem `SCS` which consists in finding a shortest common supersequence of a set of words on an alphabet Σ (NP-complete decision version denoted [SR8] in [9]). This variant considers a fixed alphabet $\Sigma = \{a_1, a_2, \ldots, a_k\}$ with $k \geq 3$, and restricts the problem `SCS` to sets of words that do not contain any word starting with letter a_1 nor any word having two identical consecutive letters. It is used in [7] and denoted `MSCS` (`Modified SCS`). In [6], Darte has shown that for any fixed $k \geq 3$, the associated decision version remains NP-complete.

We first show that k-`FI`$^{\ell}$ (with ℓ as the input threshold for the decision version of k-`FI`) is NP-complete. Let $W = \{w_1, \ldots w_m\}$ be a set of words over the alphabet $\Sigma = \{a_1, \ldots, a_k\}$. Suppose W is an instance of `MSCS`. Then, no word w_i starts with letter a_1 and thus we may construct the following k-colored reduced tree T using Σ as set of colors. T contains one root vertex r of color a_1 and m linear branches. Each branch B_i is a path of $|w_i|$ vertices that represents word w_i. Its first vertex is connected to vertex r. The p^{th} Vertex of branch B_i has color a_j if the p^{th} letter of word w_i is a_j. By the hypotheses made, the colors given to each vertex do indeed define a proper k-coloration. The root vertex r of T is chosen as the start vertex of the Flood-it game. Then, it is easy to see that a word $w = a_{\sigma_1} a_{\sigma_2} \ldots a_{\sigma_\ell}$ is a common supersequence of W, if and only if $a_{\sigma_1}, a_{\sigma_2}, \ldots, a_{\sigma_\ell}$ is a flooding sequence of colors of T.

Now, we show that the free version of the game is NP-complete. The difficulty here is that the player may choose a sequence of colors that covers all letters of all words w_i but in an order that does not respect the order according to which they appear in these words. To make sure this does not happen, we modify the tree T built in the previous paragraph. We suppose that all words in W have no more than n letters. Let T now be the tree that still has the same root vertex r as above but has $n+1$ copies of each branch B_i. On one hand, as above, any common supersequence $w = a_{\sigma_1} a_{\sigma_2} \ldots a_{\sigma_\ell}$ of W yields a flooding sequence of moves $(a_{\sigma_1}, r), (a_{\sigma_2}, r) \ldots (a_{\sigma_\ell}, r)$ that floods T entirely from vertex r only. On the other hand, the copies of each branch make sure that any optimal sequence of moves must flood from vertex r and thus yield a common supersequence of W. Indeed, suppose that $\mathcal{S}$ is an optimal flooding sequence of moves of length ℓ. Then, all copies of any branch B_i must be flooded similarly (in particular, if one of them is flooded entirely through vertex r, then they all are). Suppose one move of $\mathcal{S}$ chooses as flooding vertex the p^{th} vertex number inside a copy of branch B_i. Then, for every one of the $n+1$ copies of B_i there is a move of $\mathcal{S}$ that chooses the p^{th} vertex number of this branch as flooding vertex. Each of these $n+1$ moves floods at most two vertices (vertices $p-1$ and $p+1$ of the branch). But, since copies of B_i have no more than n vertices, $n+1$ is strictly more than the number of moves needed to flood all $n+1$ copies of B_i choosing r as flooding vertex. As a consequence $\mathcal{S}$ is not optimal. Since this is a contradiction, any optimal flooding sequence contains only moves whose flooding vertex is r. □

It is clear that any class of colored graphs whose reduced graphs contain those trees

with proper colorations, should win back the hardness results. This is formalized in the next corollary.

Corollary 1. *Let $\mathcal{G}$ be a class of graphs such that there exists a polynomial $P(.)$ and a polynomial time algorithm finding for any willow tree T with n vertices a graph $G \in \mathcal{G}$ with $\mathcal{O}(P(n))$ vertices that can be reduced to T by edge-contractions. Then for any fixed $k \geq 3$, k-FI and k-FFI are NP-hard and their decision versions are NP-complete for the class $\mathcal{G}$.*

The class of square grids (resp. triangular grids, hexagonal grids) underlying the board game with square tiles (resp. hexagonal tiles, triangular tiles) satisfies those conditions.

Figure 4 sketches the proof of the second part of Corollary 1. It pictures the shapes and the positions of the connected subsets that once contracted into single vertices give the willow tree on the left. Note that the corollary does not need to mention colorations (they are easily assigned in the hardness reduction), here grey and white zones just indicate the connected components to contract.

It yields an unified proof that the original board games Flood-It and Mad Virus are NP-hard as soon as there are 3 colors. This was unknown, but likely, for Mad Virus.

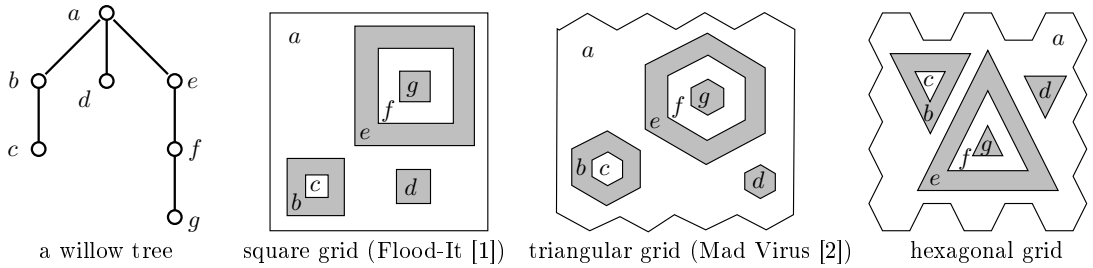

Figure 4: Willow trees can be obtained by contracting connected subsets in different kinds of grids.

Corollary 1 also implies that k-FI and k-FFI are NP-hard for $k \geq 3$ on trees of maximum degree 3. Indeed consider a willow tree T with m branches. Its root vertex may be replaced by a binary tree with m leaves (connected to the m branches of T) and with obviously no more vertices than the number of vertices of T. This yields a tree that has maximum degree 3 and that can be contracted into T.

Remark 1. The proof of Proposition 1 mixes some arguments used in the literature. First, the problem SCS is at the heart of the NP-hardness proofs about square grids [3, 14] and trees [7]. Next, the NP-hardness of k-FI is proved for willows trees in [7]. However, in this article, the complexity of the decision version of the problem does not follow because the proof relies on a new reduction from SCS to MSCS which is a Turing reduction but not a many-to-one (Karp) reduction. Moreover, their reduction can not be easily transformed into a Karp reduction since adjusting their integer decision parameter for MSCS requires some knowledge about the number of 0s in a shortest common supersequence for their initial SCS instance. The reduction used in [3] to prove the NP-completeness for k-FI on square grids is slightly different from the construction suggested in Figure 4: whereas our reduced graphs exactly fit the initial willow trees, their reduced graphs contain many

twin vertices and an extra gadjet. Nonetheless, the two reductions are close: up to quotienting by the twin relation, their reduced graphs become willow trees and the need for an additional gadjet in [3] could be avoided by using `MSCS` instead of `SCS` as starting problem. Finally the duplication argument used to prove the hardness of k-`FFI` on those trees is the same as in [3] but we show that $n+1$ copies are sufficient instead of the $2|\Sigma|n+1$ copies of [3].

Now, we show that if the maximum degree in the graph is at most 2 (paths or cycles) or the number of colors is at most 2, one can find a shortest flooding sequence in polynomial time.

Proposition 2. `FI` *on a cycle of order n can be solved in time $\mathcal{O}(n^2)$ or $\mathcal{O}(N \log N)$ where N is the number of pairs of vertices with the same color.*

Proof. The reduced graph of a k-colored cycle is still a cycle which can be computed in $\mathcal{O}(n)$. Thus we assume that we start with a proper k-coloration κ over a cycle with $V = \{0, 1, \ldots, n-1\}$ and $E = \{i(i+1) \mid 0 \leq i < n\} \cup \{(n-1)0\}$. Suppose w.l.o.g. that vertex 0 is the flooding vertex.

One can get the $\mathcal{O}(n^2)$ complexity by dynamic programming. Let us define the following variables:

$$\begin{cases} F(i,j) = \text{min nb of moves to flood } \{j, j+1, \ldots, n-1, 0, 1, \ldots, i\} & \forall\ 1 \leq i < j \leq n\text{–}1, \\ F(i,0) = \text{min nb of moves to flood } \{0, 1, \ldots, i\} & \forall\ 0 \leq i \leq n\text{–}1, \\ F(0,j) = \text{min nb of moves to flood } \{j, j+1, \ldots, n-1, 0\} & \forall\ 0 \leq j \leq n\text{–}1. \end{cases}$$

Those values can be computed thanks to the next simple recursive equations (with the convention $(n-1)+1 = 0$):

$$\begin{cases} F(0,0) = 0 & \\ F(i,0) = F(i-1,0) + 1 & \text{for all } 1 \leq i \leq n-1, \\ F(0,j) = F(0,j+1) + 1 & \text{for all } 1 \leq j \leq n-1, \\ F(i,j) = \min\{F(i-1,j)+1, F(i,j+1)+1\} & \text{if } 1 \leq i < j \leq n-1 \text{ and } \kappa(i) \neq \kappa(j), \\ F(i,j) = F(i-1,j+1)+1 & \text{if } 1 \leq i < j \leq n-1 \text{ and } \kappa(i) = \kappa(j). \end{cases}$$

The minimum length of a flooding sequence is then $\min_{0 \leq i \leq n-1} F(i, i+1)$. Its computation only requires $\mathcal{O}(n^2)$ time and one can retrieve a shortest flooding sequence with no extra cost up to a constant factor.

To get the $\mathcal{O}(N \log N)$ complexity, let us say that a couple (i,j) is *valid* if $1 \leq i < j \leq n-1$ and $\kappa(i) = \kappa(j)$. A valid couple (i,j) is said to be *interlocked* into another valid couple if (i',j') if $i < i'$ and $j' < j$. This is a partial order relation that we denote $(i,j) \prec (i',j')$. Let ℓ be the maximum length of a chain in this order, *i.e.* the maximum number of interlocked valid couples $(i_1,j_1) \prec (i_2,j_2) \prec \cdots \prec (i_\ell, j_\ell)$. Then the shortest flooding sequences have $n-1-\ell$ moves and such a sequence can be computed in $\mathcal{O}(N^2 \log N)$. Indeed, any sequence of moves propagates the flooding in both directions from vertex 0. A move floods at most two vertices since we start with a proper coloration. The sub-sequence of moves flooding two vertices simultaneously corresponds to interlocked valid couples, *i.e.* a chain for our order. The length of the

flooding sequence is clearly equal to $n-1$ minus the length of this sub-sequence (each move flooding a couple saves one move to flood the whole set $V \setminus \{0\}$). Thus, any flooding sequence has at least $n-1-\ell$ moves. Conversely, given a chain for the order $\prec$, consider the sequence of colors associated with each valid couple of the chain (in increasing order). Then one can easily complete this sequence into a flooding super-sequence where each valid couple of the chain will be flooded by only one move. By choosing a chain of maximum length ℓ one gets a shortest flooding sequence of length $n-1-\ell$.

Computing the set of the N valid couples can be done in $\mathcal{O}(N)$ by a $\mathcal{O}(n)$ bucket sort of vertices along their colors. The order $\prec$ is a partial order of dimension 2, thus one can find a chain of maximum length in $\mathcal{O}(N \log N)$ time, by using the Schensted-like algorithm presented in Knuth [10] (Section 5.1.4). □

The same reasoning and algorithm apply for paths where the flooding vertex is in any position: starting from the reduced graph which is still a path, number the vertices from one extremity of the path to the other, say that the flooding vertex is s, the same kind of dynamic programming works and for the other approach define valid couples as couples (i,j) where $i < s < j$, and define the relation $\prec$ by $(i,j) \prec (i',j')$ if $i' < i$ and $j < j'$.

Corollary 2. `FI` *on a path of order n can be solved in time $\mathcal{O}(n^2)$ or $\mathcal{O}(N \log N)$ where N is the number of pairs of vertices with the same color.*

Remark 2. Note that all paths are co-comparability graphs. A polynomial algorithm for co-comparability graphs is presented in [7]. Its complexity is not fully analyzed, but it seems that it can be implemented in $\mathcal{O}(n^2)$ for paths.

Now, we consider general graphs but with 2 colors only. As before, we work exclusively with the reduced graphs. For 2 colors, each move is only defined by the choice of a vertex s and the new reduced graph is obtained by contracting $\{s\} \cup N(s)$ into a single vertex (still denoted s). We denote this *neighborhood contraction* by $G = (V,E) \xrightarrow{s} G' = (V',E')$ where $V' = V \setminus N(s)$ and $E' = (E \setminus \{su \in E | u \in N(s)\}) \cup \{sv | \exists u \in N(x),\ uv \in E\}$. In this context, the main question is to determine the number of neighborhood contractions sufficient to reduce the initial graph to a singleton. Such a sequence of contractions is illustrated in Figure 3.

Let us recall a few definitions about distances in an undirected graph $G = (V,E)$. Let $d_G(u,v)$ be the distance between u and v in G. Vertices at distance d from u are denoted $N_G^d(u) = \{v \in V \mid d_G(u,v) = d\}$. The *eccentricity* of u is defined as $\varepsilon_G(u) = max\{d_G(u,v) \mid v \in V\}$ (which can be computed in linear time thanks to BFS from u). The *radius* of G is $R(G) = min\{\varepsilon_G(u) \mid u \in V\}$ and its *center* is $C(G) = \{u \in V \mid \varepsilon_G(u) = R(G)\}$.

The following obvious result allows to derive that 2-`FI` is in P:

Lemma 1 (folklore). *The number of neighborhood contractions at vertex s required to reduce a graph to a singleton is exactly the eccentricity $\varepsilon_G(s)$. As a consequence, solving $2-$* `FI` *comes to computing the eccentricity of the flooding vertex s in the initial reduced graph, and the shortest flooding sequence for $2-$* `FFI` *is always at most the radius of the initial reduced graph.*

PROOF. Let $G = (V, E)$ and $G' = (V', E')$ such that $G \xrightarrow{s} G'$. One can easily check that for all $d \geq 1$, $N^d_{G'}(s) = N^{d+1}_G(s)$, and thus $\varepsilon_{G'}(s) = \varepsilon_G(s) - 1$. To conclude, note that G is a singleton s if and only if $\varepsilon_G(s) = 0$. □

Now, to tackle 2-FFI, we show the intuitive result that a neighborhood contraction does not change a lot the distances (Lemma 2 and 3) and thus the radius of the graph (Lemma 4).

Lemma 2. *Let $G = (V, E)$ and $G' = (V', E')$ such that $G \xrightarrow{s} G'$. Then, $\forall u, v \in V'$, $0 \leq d_G(u, v) - d_{G'}(u, v) \leq 2$.*

PROOF. Indeed, if $d' = d_{G'}(u, v) < d = d_G(u, v)$, then, in G', there exists a path $P_{uv} = [u_0 = u, u_1, \ldots, u_{k-1}, s = u_k, u_{k+1}, \ldots, u_{d'} = v]$ from u to v passing through s of length d' (if P_{uv} didn't pass through s, it would exist in G and then $d \leq |P_{uv}| = d'$). This implies that in G, one of the two following paths of respective lengths $d' + 2$ and $d' + 1$ exists:

$$P_1 = [u, u_1, \ldots, u_{k-1}, w, s = u_k, w', u_{k+1}, \ldots, v]$$
$$\text{or } P_2 = [u, u_1, \ldots, u_{k-1}, w, w', u_{k+1}, \ldots, v]$$
$$\text{where } w \in N(s) \cap N(u_{k-1}) \text{ and } w' \in N(s) \cap N(u_{k+1}),$$

In both cases, we get $d \leq max\{|P_1|, |P_2|\} = d' + 2$. □

Lemma 3. *Let $G = (V, E)$ with radius $R(G) = R$. Then $\forall w \in C(G)$, $\forall u \in N^R(w)$, $\exists v \in N^R(w) \cup N^{R-1}(w)$ such that no path of length $d \leq R$ from w to v intersects a minimal path (of length R) from w to u.*

PROOF. The proof is illustrated by Figure 5. Define $P_{wu} = [w = u_0, \ldots, u_R = u]$ as a minimal path from w to u. Note that the following necessarily holds: $d_G(u_1, u) = R - 1$ and $\forall v \in V$, $d_G(u_1, v) \leq 1 + d_G(w, v)$. If there is no $v \neq u$ in $N^R(w) \cup N^{R-1}(w)$, then $\varepsilon(u_1) = R - 1 < R$ which is a contradiction. Thus, let $v \in N^R(w) \cup N^{R-1}(w)$, $v \neq u$ and let $P_{wv} = [w = v_0, \ldots, v_d = v]$, be a path of length $d \leq R$ from w to v. If P_{wv} intersects P_{wu}, then there exists i and j such that $u_i = v_j$ and $[u_1, u_2, \ldots, u_i = v_j, \ldots, v_d = v]$ is a path of length no greater than $d - 1 \leq R - 1$ ($i \leq j$ because otherwise there is a path from w to u shorter than P_{wu}) from u_1 to v. As a consequence, $d_G(u_1, v) \leq R - 1$. Thus, again, if forall vertex v in $N^R(w) \cup N^{R-1}(w)$, there is a path of length $d \leq R$ from w to v intersecting a minimal path from w to u, then $\varepsilon(u_1) = R - 1 < R$. This is contradiction. □

Lemma 4. *Let $G = (V, E)$ and $G' = (V', E')$ such that $G \xrightarrow{s} G'$. Then, $R(G) - 1 \leq R(G') \leq R(G)$. Moreover, if $s \in C(G)$, then $R(G') = R(G) - 1$ and $s \in C(G')$.*

PROOF. Trivially, $R(G')$ cannot be bigger than $R(G) = R$. Let us show that it cannot be smaller than $R - 1$. Assume that it is: $R(G') \leq R - 2$. Let $w \in C(G')$. By hypothesis, $\forall v \in V$, $d_{G'}(w, v) \leq R - 2$. By Lemma 2, it also holds that $\forall v \in V$, $d_G(w, v) - d_{G'}(w, v) \leq 2$. Consequently, $\forall v \in V$, $d_G(w, v) \leq R$, *i.e.*, $w \in C(G)$. Now, let $u \in N^R(w)$ and v be

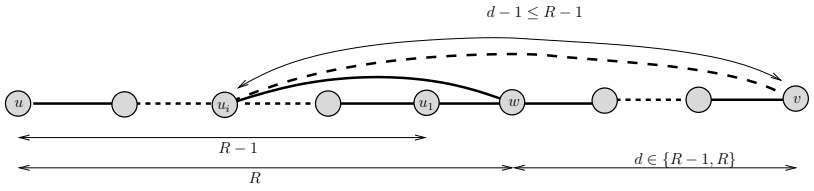

Figure 5: Configuration of paths in the proof of Lemma 3

as in Lemma 3: $d_G(w,u) = R$, $d_G(w,v) \in \{R-1, R\}$ and no path of length $d \leq R$ from w to v intersects a minimal path from w to u.

On the first hand, since $d_G(w,u) = R$, $R(G') \leq R-2$ and $d_G(w,u) - d_{G'}(w,u) \geq 2$, the following holds: $d_G(w,u) - d_{G'}(w,u) = 2$. Therefore, in G, any minimal path P_{wu} from w to u passes through s. On the other hand, for similar reasons, $d_G(w,v) - d_{G'}(w,v) \geq 1$ but no minimal path P_{wv} (of length $d_G(w,v) \in \{R-1, R\}$) from w to v passes through s (otherwise it would intersect P_{wu}). However, since $d_{G'}(w,u) < d_G(w,u)$, P_{wv} must be shortened in G'. From this, we derive that $d_G(w,v) - d_{G'}(w,v) = 1$, $d_G(w,v) = R-1$ and P_{wv} passes through exactly 2 neighbors s_1 and s_2 of s: $P_{wv} = [w = v_0, \ldots, s_1 = v_i, s_2 = v_{i+1}, \ldots, v_d = v]$. But then, $[w = v_0, \ldots, s_1 = v_i, s, s_2 = v_{i+1}, \ldots, v_{R-1} = v]$ is a path of length $(R-1)+1 = R$ from w to v that intersects P_{wu}. This is a contradiction.

The statement about neighborhood contractions at the center is a direct consequence of $\varepsilon_{G'}(s) = \varepsilon_G(s) - 1$ already noticed for Lemma 1. □

As a consequence of Lemma 1 and 4, we get:

Proposition 3. *A graph $G = (V, E)$ can be reduced to a singleton by $R(G)$ neighborhood contractions and no less. To do so, one can choose any vertex in $C(G)$ to perform all the contractions.*

Proof. By Lemma 1, it can be done in $R(G)$ moves and Lemma 4 ensures it cannot be done in less. □

Note that the proposition and all the previous lemmas apply to any connected graph, not necessarily bipartite (as the reduced graph of 2-`FFI`). Now if we come back to 2-`FFI`, since the reduced graph of an arbitrary colored graph may be constructed in linear time, by applying Proposition 3 reinterpreted in terms of flooding sequence, we settle the complexity of the puzzle (problem left open in [3] for $n \times n$ square boards):

Proposition 4. 2-`FFI` *is in P: it can be solved in time $\mathcal{O}(nm)$, i.e., the time needed to compute the radius of the associated reduced graph (and to find a vertex in the center of this graph).*

Remark 3. Two different and independent proofs have been provided in [5] and [12]. In [5], the proof is stated for $n \times n$ square boards, but a careful look shows that the arguments apply for any graph.

3. Conclusion

Though board configurations have a particular structure, *e.g.* their reduced graphs are planar, we have showed that studying the board flooding games with a graph point of view is valuable. It has provided a proof that 2-FFI is polynomial by putting forward the core parameter which is the radius of a graph.

Nonetheless, one might wish to exploit the characteristics of $n \times n$ colored square boards. For instance, is it possible to find faster algorithms to compute the radius of their reduced graphs ?

More generally, few classes of graphs have been studied and it would be interesting to unveil general features which imply polynomial or NP-hard complexities. In particular which graph parameters ensure polynomial solutions, once bounded ?

References

[1] Flood-it (flash game), http://floodit.appspot.com/, 2011.

[2] Mad virus (flash game), http://www.bubblebox.com/play/puzzle/539.htm, 2011.

[3] D. Arthur, R. Clifford, M. Jalsenius, A. Montanaro, B. Sach, The complexity of flood filling games, in: FUN'2010, volume 6099 of *LNCS*, Springer, June 2010, pp. 307–318.

[4] A. Born, Flash application for the computer game "biene" (honey-bee), http://www.ursulinen.asn-graz.ac.at/Bugs/htm/games/biene.htm, 2009.

[5] R. Clifford, M. Jalsenius, A. Montanaro, B. Sach, The complexity of flood filling games, arXiv:1001.4420v2 [cs.DS], August 2010. To appear in Theory of Computing Systems.

[6] A. Darte, On the complexity of loop fusion, Parallel Computing 26 (2000) 1175–1193.

[7] R. Fleischer, G.J. Woeginger, An algorithmic analysis of the honey-bee game, in: FUN'2010, volume 6099 of *LNCS*, Springer, August 2010, pp. 178–189.

[8] H. Fukui, A. Nakanishi, R. Uehara, T. Uno, Y. Uno, The complexity of free flood filling games, Information Processing Society of Japan (IPSJ) SIG Notes 2011 (August 2011) 1–5.

[9] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, 1979.

[10] D.E. Knuth, The Art of Computer Programming, volume 3, Addison-Wesley, 1973. Section 5.1.4.

[11] A. Lagoutte, Jeux d'inondation dans les graphes, HAL archive hal-00509488, August 2010. In french.

[12] K. Meeks, A. Scott, The complexity of flood-filling games on graphs, arXiv:1101.5876v1 [cs.DS], January 2011. To appear in Discrete Applied Maths.

[13] K. Meeks, A. Scott, The complexity of free-flood-it on $2 \times n$ boards, arXiv:1101.5518v1 [cs.DS], January 2011.

[14] E. Verbin, Is this game np-hard?, Sariel blog, May 2009. http://valis.cs.uiuc.edu/blog/?p=2005.