RMIT UNIVERSITY

Cops, Robbers and Firefighters on Graphs

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

David Ellison

Masters in Mathematics, Ecole Polytechnique, France

School of Science

College of Science, Engineering and Health

RMIT University

July 2018

RMIT UNIVERSITY

Declaration for candidates submitting a thesis

I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; any editorial work, paid or unpaid, carried out by a third party is acknowledged; and, ethics procedures and guidelines have been followed.

16.07.2018

David Ellison

Contents

List of Figures

Abstract

This thesis focuses on the game of cops and robbers on graphs, which was introduced independently by Quilliot in 1978 and by Nowakowski and Winkler in 1983, and one of its variants, the firefighter problem. In the game of cops and robbers, the cops start by choosing their starting positions on vertices of a graph, then the robber chooses his starting point. Then, they move each in turn along the edges of the graph. The basic objective is to determine whether the cops have a strategy which allows them to catch the robber. Looped vertices allow the cops and the robber to pass their turn. The first chapter explores the effect of loops on the cop number and the capture time. It provides examples of graphs where the cop number almost doubles when the loops are removed, graphs where the cop number decreases when the loops are removed, graphs where the capture time is quadratic in the number of vertices and copwin graphs where the cop needs to move away from the robber in optimal play.

In the second chapter, we investigate the links between this game and algebraic topology. We extend the game of cops and robbers on graphs by considering the case where the cops chase the image of the robber by a graph homomorphism. We prove that the cop number associated with a graph homomorphism is a homotopic invariant. Homotopies between graph homomorphisms or homotopy equivalences between graphs allow us to compare their cop numbers and also their capture times. Finally, using homotopic invariants such as homology groups, we investigate structural properties of copwin graphs.

Finally, in the third chapter, we explore the FIREFIGHTER problem, introduced by Hartnell in 1995, where a fire spreads through a graph while a player chooses which vertices to protect in order to contain it. While focusing on the case of trees, we also consider a variant game called FRACTIONAL FIREFIGHTER in which the amount of protection allocated to a vertex lies between 0 and 1. While most of the work in this area deals with a constant amount of firefighters available at each turn, we consider three research questions which arise when including the sequence of firefighters as part of the instance. We first introduce an online version of both FIREFIGHTER and FRACTIONAL FIREFIGHTER, in which the number of firefighters available at each turn is revealed over time. We show that a greedy algorithm on finite trees is 1/2-competitive for both online versions, which generalises a result previously known for special cases of FIREFIGHTER. We also show that the optimal competitive ratio of online FIREFIGHTER ranges between 1/2 and the inverse of the golden ratio. Next, given two firefighter sequences, we discuss sufficient conditions for the existence of an infinite tree that *separates* them, in the sense that the fire can be contained with one sequence but not with the other. To this aim, we study a new purely numerical game called *targeting game*. Finally, we give sufficient conditions for the fire to be contained on infinite trees, expressed as the asymptotic comparison of the number of firefighters and the size of the tree levels.

Chapter 0

Introduction

Like linear programming, information theory and many other areas, the study of pursuit-evasion games was initially motivated by the military conflict of WWII. The mathematical study of these games was first introduced in the early fifties by Rufus Isaacs [13], with an application to missile guidance systems. The first pursuit-evasion games on graphs were introduced by Torrence Parsons in 1976 [48], and the study of pursuit-evasion games was then split between continuous games and discrete games. The game of cops and robbers was introduced only two years later by Alain Quilliot [49] and independently by Richard Nowakowski and Peter Winkler in 1983 [47]. Since then, the game of cops and robbers on graphs has become the prime example of a discrete pursuit-evasion game and is now the subject of a wide literature.

The game of cops and robbers on graphs is the starting point of this thesis. In this game, a robber and one or several cops move each in turn along the edges of a graph, and the basic objective is to work out whether the cops can catch the robber. This game has quite a few applications. The first and most important, theoretically speaking, is network security: a network may be considered more secure if fewer cops are needed to patrol it. There are also applications to computer viruses as well as biological viruses. Another application is to the problem of pipe networks in North America which are contaminated by regenerating agents; that is, either algae or zebra mussels. The pipes are cleansed using automated robots, and determining the minimum number of robots required to fight the infection, as well as their optimal path along the network, is a direct application of a variant of the game. Many such variants of the game of cops and robbers have been studied: some were created through slight changes in the rules, like modifying the victory condition, playing with incomplete information, disallowing the players to pass their turn, speeding up the robber or allowing the cops to use helicopters (see [10] for a survey of these variant games). Other variants involve significantly different objectives, like seepage [18], guarding [29], searching [51] or sweeping [54]. The firefighter variant has given rise to a wide literature; it is also the topic of Chapter 3 in this thesis.

Our study of the firefighter problem began as a collaboration with Bertrand Jouve and Pierre Coupechoux for the project Geosafe H2020. Geosafe is a joint project between Australia and Europe, the purpose of which is to make our planet safer before year 2020 by addressing the threat of bushfires. The firefighter problem is a variant game derived from the game of cops and robbers in which a fire spreads through a graph and the firefighter tries to contain it, to the best of his abilities, by choosing at each turn which vertices to protect. At each turn, the fire spreads to all adjacent unprotected vertices. Other people working for Geosafe modelise the fire spread by taking into account weather, vegetation, topography, smoke front, ember spray, roads, electric lines and many other factors; but our simplistic approach already leads to very complex and interesting problems. Rather than trying to resolve practical issues, our aim is to further our general understanding of percolation problems on graphs.

The first two chapters of this thesis are dedicated to the game of cops and robbers while the third focuses on the firefighter problem. Although these two games are similar in nature, they led to very different types of study. The first chapter is mostly game theory: it covers the effect of a small change in the rules of cop and robbers and analyses playing strategies which apply in specific examples. The second chapter combines cops and robbers with algebraic topology, starting from the use of homotopies to characterise copwin graphs. The study of the firefighter problem in the third chapter focuses on approximation results and online algorithms as well as specific playing strategies.

Notations and definitions

We now introduce some notations and standard definitions of set theory and graph theory, which will be used throughout the thesis. Notations and definitions which are specific to a chapter will be specified in the related chapter.

Given two sets A and B, let $A \setminus B = \{x \in A, x \notin B\}$. Let $\emptyset$ denote the *empty set*; $\mathbb{N}$ will denote the *set of non-negative integers* and $\mathbb{N}^* = \mathbb{N} \setminus \{0\}$; $\mathbb{Z}$ will denote the *set of integers*, $\mathbb{R}$ will denote the *set of real numbers* and $\mathbb{R}^+$ the *set of non-negative real numbers*. The *cardinality* of A will be denoted $|A|$. The set of functions from B to A is denoted A^B, in particular, $A^{\mathbb{N}}$ denotes the set of sequences of A. Given $i, i' \in \mathbb{N}$, $\mathrm{lcm}(i, i')$ and $\gcd(i, i')$ will respectively denote the *lowest common multiple* and the *greatest common divisor* of i and i'. Given a predicate P, we denote by $\mathbb{1}_P$ the associated *characteristic function* so that $\mathbb{1}_{P(x)} = 1$ if $P(x)$ is true and 0 otherwise. Given a linear map f, $Im\ f$ and $Ker\ f$ denote its *image* and *kernel*, respectively.

Given a vertex u of a graph $G = (V, E)$, we denote by $N(u)$ and $N[u]$ the *open* and *closed neighbourhoods* of u respectively ($N[u] = N(u) \cup \{u\}$). Note that the open and closed neighbourhoods of a vertex u are equal if there is a loop on u, hence the distinction vanishes entirely in the case of totally looped graphs. The *Cartesian product* of two graphs $X = (V, E)$ and $Y = (V', E')$, denoted $X \square Y$, is the graph with vertex set $V \times V'$ and edges between (x_1, y_1) and (x_2, y_2) if $x_1 = x_2$ and $y_1 \in N(y_2)$ or if $y_1 = y_2$ and $x_1 \in N(x_2)$. Using the definition in [52], the *categorical product* of two graphs $X = (V, E)$ and $Y = (V', E')$, denoted $X \times Y$, is the graph with vertex set $V \times V'$ and edges between (x_1, y_1) and (x_2, y_2) if $x_1 \in N[x_2]$ and $y_1 \in N[y_2]$. Let P_n denote the *path* with n vertices numbered 0 to $n - 1$. Let $Hom(X, Y)$ denote the *set of graph homomorphisms* from X to Y, Id_X the *identity* of X, and $Aut(X)$ the *group of automorphisms* of X.

A *tree* T is a connected graph without cycles; $V(T)$ and $E(T)$ will denote its vertex set and edge set, respectively. A *rooted tree* is a pair (T, r), where $r \in V(T)$ is the *root*. The *leaves* of a tree are its vertices of degree 1, excluding the root, unless the root is the only vertex, in which case it is counted as a leaf; the other vertices are called *internal vertices*. Vertices in a rooted tree are referred to as *parent*, *sibling*, *child*, *ancestor* or *descendant* of another vertex, depending on their respective positions in relation to the root. Given an integer $p \geq 2$, a *perfect p-ary tree* is a tree where every internal vertex has p children and the leaves are all at the same distance from the root. An $n \times m$ *grid* is the graph $P_n \square P_m$. The $n \times m$ torus grid is the Cartesian product of two cycles of lengths n and m. The Hamming graph $H(d, q)$ has vertex set $\{0, \ldots, q-1\}^d$, with its vertices adjacent if and only if they differ in exactly one coordinate. Given a graph G, a *clique* is a set of vertices of G which are all pairwise adjacent and a *stable set* is a set of vertices with no edges between them. An induced path of length n of G is a sequence of vertices $(u_0, \ldots, u_n)$ where u_i and u_{i+1} are adjacent and no vertex is repeated.

When in doubt, the reader is referred to [21] for all standard vocabulary of graph theory, to [31] for complexity theory, to [38] for algebraic topology and to [10] for the topic of cops and robbers.

Chapter 1

The Impact of Loops on the Game of Cops and Robbers on Graphs

1.1 Introduction

The Game of Cops and Robbers is a two-player game played on a graph G. In this game, the cop starts by choosing her starting position on a vertex of G. Then, the robber chooses his starting point. Following this, they move each in turn along the edges of the graph. We assume that they play with complete information, meaning that they can see each other at all times. The cop's objective is to catch the robber, which she can achieve either by moving onto the robber's position, or by forcing him to move onto her position. The question is: does the cop have a winning strategy? If so, the graph is said to be *copwin*, otherwise it is *robberwin*. When the graph is robberwin, the question then becomes: how many cops do you need to catch the robber? When playing with several cops, all the cops must move during their turn. However, vertices with a loop allow a subset of the cops or the robber to stay in place by following the loop. The minimum number of cops required to catch a robber on a graph G is called the *cop number* of G and is denoted $c(G)$. The notion of cop number was first introduced in the seminal paper by Aigner and Fromme in 1984 [2].

Most authors play on totally looped graphs, or consider only graphs without loops and allow the cops and the robber to pass their turn whenever they wish to do so.

Note that some authors, e.g. [14], consider moving onto the robber's position the only way for the cop to win. In that case, if there are no loops, the robber will win by shadowing her movements. Their version has one advantage: copwin graphs become easy to recognise using dismantlability or other equivalent criteria (see theorem 4.1 in [14]). However, our winning condition seems more natural and the difficulties which arise from no longer having an easy characterisation of copwin graphs lead to interesting possibilities, which will be highlighted in this chapter.

It is always possible to replace a looped vertex by a pair of adjacent twin vertices; hence, any cops and robber problem on a totally looped or partially looped graph can be reduced to a similar problem on a simple graph.

We consider that the cops' objective is to catch the robber as swiftly as possible, whereas the robber tries to evade the cops for as long as possible. Hence, optimal play for the cops involves minimizing the capture time (introduced in [8]) in the worst case scenario, while an optimal strategy for the robber maximises the same capture time.

Given a graph G, let G^+ and G^- denote the same graph with loops respectively added or removed on every vertex.

Previous authors already considered the effect of allowing or forbidding the players to pass their turn: in [46], Neufeld and Nowakowski distinguish between the *passive game*, where both sides may pass their turn, and the *active game*, where the robber and at least one cop must move during their turn. While the former is equivalent to playing on totally looped graphs with our definition, the latter is different from playing without loops with our definition, and matches only when there is a single cop. In [12] (Lemma 16), it is shown that $c(G^+) \leq c(G^-) + 1$, which means that the loops can only be slightly of help to the robber. In this chapter, we will provide examples where this bound is reached for any cop number. We will also provide examples where the cop number almost doubles when the loops are removed instead.

In [12], Boyer et al. also disprove a conjecture which states that on copwin graphs, in optimal play, the distance between the cop and the robber is decreasing. In their counter-example, the cop has to make a move which keeps her at the same distance from the robber, and he then moves away from her. We provide examples of graphs where the cop actually moves away from the robber in optimal play, both in the totally looped and partially looped cases.

1.2 Comparing the cop numbers of totally looped and non-looped graphs

A vertex u is said to be *dismantlable* if there exists a vertex $v \neq u$ such that $N(u) \subset N[v]$. Vertex u is then said to be *dominated* by v. This means that if the robber is standing on u and the cop on v, and if it is the robber's turn to move, then he will be caught on the cop's next move. The robber is then said to be *cornered*. We say that a cop on v *covers* the vertices in $N[v]$ in the sense that she prevents the robber from moving to these vertices. A totally looped graph is said to be *dismantlable* if it can be reduced to a single vertex by successively removing dominated vertices. Quilliot[49] and Nowakowski and Winkler[47] proved independently that a totally looped graph is copwin if and only if it is dismantlable.

If a graph G has several connected components $G_1, \ldots, G_m$, then we have $c(G) = \sum_{i=1}^{m} c(G_i)$. Hence, from now on, we will consider only connected graphs.

Proposition 1. *For any connected graph G, $c(G^-) \leq 2c(G^+)$.*

Proof. Given a winning strategy for p cops on G^+, for each cop, we can place another cop on a neighbouring vertex in the initial setup. In G^-, the extra cop will shadow the former and switch with her should she need to remain in place in the original winning strategy on G^+. $\square$

Remark 1. *It is shown in [12] that $\forall G$, $c(G^+) \leq c(G^-) + 1$. The idea is that one cop should follow the robber while the others apply their strategy for G^-. If the robber follows a* *loop, the cop following him gets closer while the others pass their turn. Thus we obtain that* $\forall G, c(G^+) - 1 \leq c(G^-) \leq 2c(G^+)$.

Remark 2. *For any connected graph G, not necessarily without loops, we have $c(G^+) \leq c(G) + 1$. Indeed, if G is partially looped, when the robber passes his turn, the cops applying the strategy for G can go back to their initial positions and start their strategy anew. The proof of proposition 1 also shows that even if we allow the robber to pass his turn while the cops may not, then the cop number would be at most $2c(G^+)$.*

Proposition 2. *For all $p \in \mathbb{N}^*$, there exists a connected graph H_p such that $c(H_p^+) = p + 1$ and $c(H_p^-) = p$.*

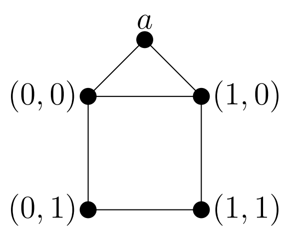


Figure 1.1: Graph H_1

Proof. Let H_p be the graph obtained by adding to the Hamming graph $H(2p, 2)$ a vertex a linked to vertices $(0, \ldots, 0)$ and $(1, 0, \ldots, 0)$. Since H_p is non-looped, $H_p = H_p^-$. The degree of each vertex in $H(2p, 2)$ is $2p$, and any two vertices at distance 2 share exactly 2 common neighbours. Also, adjacent vertices have disjoint open neighbourhoods. Hence, while p cops may cover the open neighbourhood of any given vertex in $H(2p, 2)$, they cannot cover its closed neighbourhood. This means that on $H(2p, 2)^+$, the robber has a winning strategy against p cops. After adding vertex a, the same strategy still works since a is dominated by $(0, \ldots, 0)$, so whenever a cop goes to a, the robber can pretend that she went to $(0, \ldots, 0)$ instead. Therefore, $c(H_p^+) > p$.

We will now show by induction that on H_p, p cops have a winning strategy which satisfies the following properties:

- all the cops start on vertex a,
- all the cops place themselves at an even distance from the robber after each move,
- the robber is prevented from ever reaching a.

On the graph H_1, there exists such a strategy that allows a single cop to win in two moves. Let us now assume that we have such a strategy for p cops on H_p, for a fixed p. On H_{p+1}, the strategy for $p + 1$ cops may be divided into the following steps:

1. First, all $p + 1$ cops start at a. The robber then chooses his starting point.

2. Since $H(2p+2,2)$ is bipartite, either $(0,\dots,0)$ or $(1,0,\dots,0)$ is at an even distance from the robber's starting point. The cops all move to that one. The bipartite nature of $H(2p+2,2)$ guarantees that the parity of the distance will remain even after the cops' move throughout the game, provided that no one goes through a.

3. This step applies only if the robber's first move brings him to a vertex where the last two coordinates are either $(0,1)$ or $(1,0)$. In this case, the first p cops will move to match their last two coordinates with the robber's. In all cases, the first p cops now have an even number of coordinates which differ from the robber's, both among the first $2p$ coordinates and among the last two.

4. From this point on, by omitting the last two coordinates, the first p cops apply the strategy for H_p; unless the robber moves along one of the last two coordinates, in which case they move to match their last two coordinates with the robber's. As a result, either the robber's last two coordinates remain $(1,1)$ and the cops' $(0,0)$, or those p cops and the robber have matching last two coordinates. This means that the robber is prevented from reaching vertex a.

5. Meanwhile, the last cop will always move along the first of her coordinates which differ from the robber's. Thus, the last cop will catch the robber if he moves along the last two coordinates more than $2p$ times. If not, the strategy for H_p will lead to one of the cops aligning at least all but the last two of her coordinates with the robber's. This means that either the robber is caught, or his last two coordinates have remained $(1,1)$ until now.

6. In that case, the cop who almost caught the robber will then maintain the alignment of the first $2p$ coordinates throughout the rest of the game, thus preventing the robber from making any further movements along the last two coordinates without immediately getting caught.

7. The other p cops now go to vertex $(0,\dots,0,1,1)$.

8. Those p cops apply again the strategy for H_p starting from the second move, which results in the robber's capture.

Hence, $\forall p$, $c(H_p) \leq p$, and it follows from proposition 1 that we have $c(H_p) = c(H_p^-) = p$ and $c(H_p^+) = p+1$.

□

We will now construct a sequence of graphs $G_p, p \geq 2$ such that $c(G_p^+) = p$ and $c(G_p^-) = 2p-1$. For $p \geq 2$, let G_p be the following graph consisting of three layers. In order to build the first layer, take a perfect $4p^2$-ary tree of height 11 and join the children of each internal vertex into $2p$ cliques of $2p$ vertices. Thus, other than the root and leaves, every vertex has 1 parent, $2p-1$ adjacent siblings and $4p^2$ children. Note that this graph is copwin, with or without loops, as the cop wins by always moving towards the robber. We now number the vertices by level, from 1 at the root down to the leaves, so that sibling vertices have consecutive numbers. For all $(2p)^{13} \leq i < (2p)^{13} + p^2$ and $0 \leq a < i$, we add a vertex $v_{i,a}$ on layer 2 linked to every vertex of layer 1 with a number congruent to a mod i. Hence, no two sibling vertices (adjacent or non-adjacent) are linked to the same $v_{i,a}$. Layer 3 contains $p-1$ vertices $v_1, \ldots, v_{p-1}$, where v_k is linked to $v_{i,a}$ iff $i \equiv k \mod (p-1)$. Since G_p is non-looped, $G_p = G_p^-$.

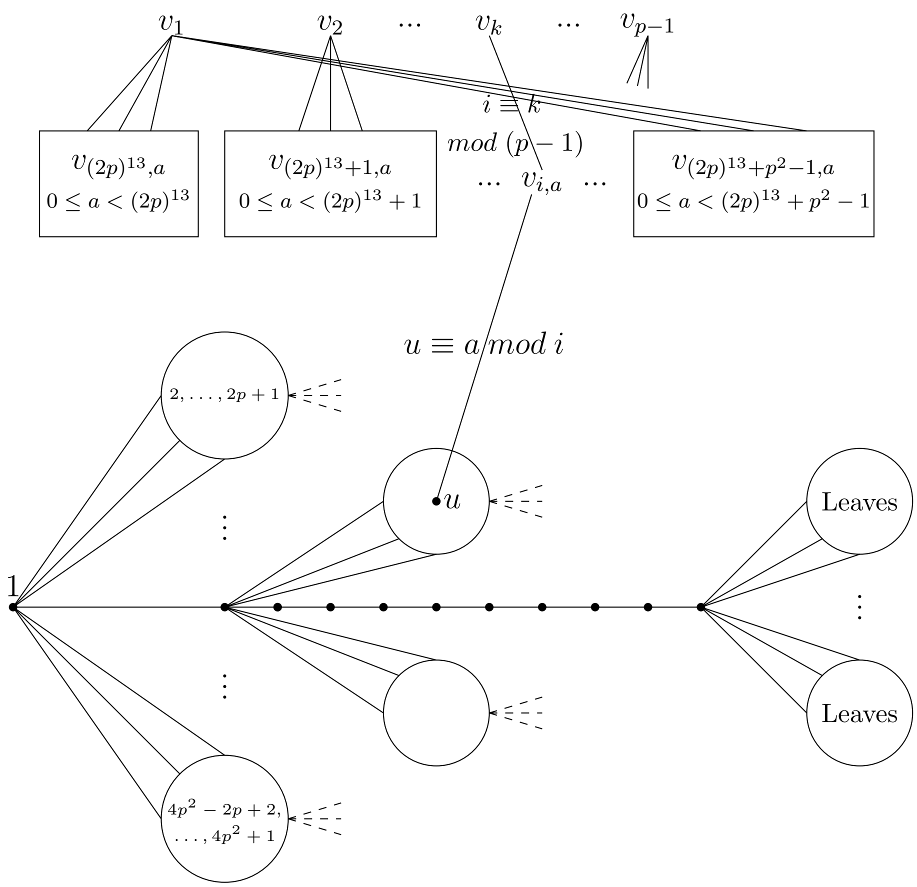


Figure 1.2: The three layers of G_p: circles represent cliques, rectangles are stables.

While the perfect tree we used as a base for G_p is, strictly speaking, no longer a tree, since edges were added to it, we will keep using the vocabulary of trees, in particular "leaf" and "internal vertex", as a convenient way to refer to its vertices and their relative positions. We will also identify a vertex u in layer 1 with its number $1 \leq u \leq \sum_{k=0}^{11}(2p)^{2k}$.

The construction of the graph G_p underwent many changes throughout four years of research. The main idea was to create a graph with key vertices which need to be occupied at all times in order to catch the robber efficiently. Despite the simplicity of this concept, the end result is a very complex structure. Initial versions for G_2 used a grid with diagonals instead of a tree and had routes via layer 2 allowing the robber to escape to other quadrants of the grid. However, proving the properties is impractical and it therefore remains uncertain whether these graphs are actually valid examples or not. In order to reach the current setup, we used the following criteria in order to find working parameters: choose a perfect n-ary tree of height h, where sibling vertices are grouped in d cliques of size c, with M internal vertices and N vertices in total. In layer 2, $i_{min} \leq i < i_{max}$ and $0 \leq a < i$. The proof requires the following constraints:

- $i_{min} \geq n$ so that no two siblings are adjacent to the same $v_{i,a}$.
- $d > 2p-2$ so that the robber cannot be trapped on an internal vertex (lemma 2, point 1).
- $M > 3(2p-2)i_{max}$ so that the robber can always go from $v_{i,a}$ to an internal vertex of layer 1 (lemma 2, point 2).
- $i_{min} \geq 2p-2$ so that the robber cannot be trapped on v_k (lemma 2, point 3).
- $i_{min}^2 > N(i_{max}-i_{min})$ so that two vertices of layer 1 cannot have more than one common neighbour in layer 2 (lemma 3).
- $i_{max} - i_{min} > (p-1)(p+1)$ for the conclusion of lemma 3.
- $c \geq p+1$ so that $p-1$ cops on layer 2 and $p-1$ cops on layer 3, alternating at each turn, cannot trap the robber (lemma 4).

The values selected for G_p are among the simplest that satisfy all the above conditions for all values of p.

Lemma 1. *For all $p \geq 2$, $c(G_p^+) \leq p$ and $c(G_p^-) \leq 2p-1$.*

Proof. A winning strategy for p cops on G_p^+ consists of having $p-1$ cops occupy the vertices of layer 3, thus denying access to layer 2 to the robber, while the last cop chases the robber on layer 1, which is copwin. Similarly, $2p-1$ cops can catch the robber on G_p by having $p-1$ pairs alternating between layers 2 and 3, thus always occupying layer 3, while the last cop chases the robber on layer 1, eventually cornering him on a leaf. □

Lemma 2. *For all $p \geq 2$, the only vertices of G_p where the robber may be cornered by $2p-2$ cops are the leaves of layer 1.*

Proof. In order to prove this, we need to consider three cases: the internal vertices of layer 1, all of layer 2 and layer 3.

1. An internal vertex of layer 1 has $4p^2$ children, sorted into $2p$ cliques. No other vertex in the graph has neighbours in two of these cliques. Hence, at least $2p$ cops are needed to trap the robber on an internal vertex of layer 1.
2. Let M denote the number of internal vertices on layer 1: $M = \sum_{k=0}^{10}(2p)^{2k}$. For any vertex x, let $N_{iv}(x)$ denote the intersection of $N(x)$ with the internal vertices of layer 1. Given any vertex $v_{i,a}$ of layer 2, no two elements of $N_{iv}(v_{i,a})$ are siblings. Hence, a cop in layer 1 will cover at most 3 elements of $N_{iv}(v_{i,a})$ (the parent, one child and either a sibling or her own position). Also, for $(i',a') \neq (i,a)$,

$$|N_{iv}(v_{i,a}) \cap N_{iv}(v_{i',a'})| \in \{0, \left\lfloor \frac{M}{\mathrm{lcm}(i,i')} \right\rfloor, \left\lceil \frac{M}{\mathrm{lcm}(i,i')} \right\rceil\},$$

depending on the values of a and a'. For $i \neq i'$, since $\gcd(i,i') \leq |i-i'| \leq p^2$, we have $\operatorname{lcm}(i,i') = \frac{ii'}{\gcd(i,i')} \geq 2^{26}p^{24} > M$. Thus,

$$|N_{iv}(v_{i,a}) \cap N_{iv}(v_{i',a'})| \leq 1.$$

Finally, for v_k in layer 3, $N_{iv}(v_k) = \emptyset$. On the whole:

$$\forall x \in G,\ x \neq v_{i,a},\ |N_{iv}(v_{i,a}) \cap N_{iv}(x)| \leq 3.$$

The cardinality of $N_{iv}(v_{i,a})$ is either $\lceil \frac{M}{i} \rceil$ or $\lfloor \frac{M}{i} \rfloor$. Hence,

$$|N_{iv}(v_{i,a})| \geq \left\lfloor \frac{M}{(2p)^{13}+p^2} \right\rfloor > 3(2p-2).$$

This means that if the robber is in layer 2, $2p-2$ cops cannot cover all the internal vertices of layer 1 in his neighbourhood.

3. Let us now consider vertex v_k of layer 3. Its neighbours are the $v_{i,a}$ with $i \equiv k \mod (p-1)$. For such an i and $a \neq a'$, no vertex other than v_k is adjacent to both $v_{i,a}$ and $v_{i,a'}$. Since $i > 2p-2$, it is impossible to trap the robber on layer 3 with only $2p-2$ cops.

Thus, the only vertices of G_p where the robber might be cornered by $2p-2$ cops are the leaves of layer 1. □

Lemma 3. *For all $p \geq 2$, with $2p-2$ cops, if the robber is on a vertex of layer 1 of G_p, he can always move to layer 2 without getting immediately caught, unless all vertices of layer 3 are occupied by cops.*

Proof. For u in layer 1, let $N_{L2}(u) = \{v_{i,a},\ u \equiv a \mod i\}$.

$$\forall u' \neq u, |N_{L2}(u) \cap N_{L2}(u')| = |\{(2p)^{13} \leq i < (2p)^{13}+p^2, i|u-u'\}|.$$

For $(2p)^{13} \leq i,i' < (2p)^{13}+p^2$, $i \neq i'$, we have $\operatorname{lcm}(i,i') \geq 2^{26}p^{24}$. Since layer 1 has fewer than $2^{26}p^{24}$ vertices, this means that there cannot be two distinct i and i' which divide $u-u'$. Hence:

$$\forall u' \neq u, |N_{L2}(u) \cap N_{L2}(u')| \leq 1.$$

Note that for any vertex u in layer 1 and any v_k in layer 3,

$$\begin{aligned}|N_{L2}(u) \cap N(v_k)| &= |\{v_{i,a},\ u \equiv a \mod i, i \equiv k \mod (p-1)\}| \\ &= |\{(2p)^{13} \leq i < (2p)^{13}+p^2,\ i \equiv k \mod (p-1)\}|.\end{aligned}$$

Thus, $|N_{L2}(u) \cap N(v_k)| = p+1$ for all but one value $2 \leq k \leq p-1$ and $\exists! k$, $2 \leq k \leq p-1$, $|N_{L2}(u) \cap N(v_k)| = p+2$. It follows that if the robber is in layer 1 and one or several vertices of layer 3 are not occupied by cops, the remaining cops are insufficient to prevent the robber from reaching layer 2. □

Lemma 4. *For all $p \geq 2$, the robber can evade $2p-2$ cops on G_p.*

Proof. The robber's winning strategy against $2p-2$ cops can be simply described as follows: choose a starting position on a grandchild of the root at distance at least 2 from every cop, then, move arbitrarily while remaining at distance 2 from the cops and avoid moving to the leaves of layer 1 unless forced to do so. Finding a suitable starting position is always possible since at least one child of the root is not initially occupied by a cop, and it follows from the proof of lemma 2 part 1 that at least one of its children is not covered by the cops. We proved in lemma 2 that if the robber is in layer 2, he can always escape to an internal vertex of layer 1. Note that if there are $p-1$ cops on layer 3 and $p-1$ cops on layer 2, the robber can always move to an adjacent sibling vertex since at most $p-1$ of those are covered by the cops. Hence, it follows from lemma 3 that the only situation where the cops can force the robber to move to a leaf is if the robber is on the parent of that leaf, $p-1$ cops occupy layer 3 and one cop occupies the parent of the robber's vertex. In this situation, after the robber moves to a leaf, the cops on layer 3 have to move to layer 2, and at most $p-2$ cops can move to layer 3. Hence, the robber will be able to escape via layer 2 immediately after being forced to move to a leaf of layer 1. □

We can now conclude that G_p indeed has the required property, which gives us the following proposition:

Proposition 3. *For all $p \geq 2$, there exists a connected graph X_p such that $c(X_p^+) = p$ and $c(X_p^-) = 2p-1$.*

Proof. The graph G_p has this property. Indeed, it stems from lemma 1 and lemma 4 that $c(G_p) = c(G_p^-) = 2p-1$, and it follows from lemma 1 and proposition 1 that $c(G_p^+) = p$. □

Proposition 1 states that removing all the loops from a totally looped graph at most doubles the cop number. We have shown that removing all the loops from G_p^+ almost doubles the cop number. Not only is this asymptotically optimal, we believe that this is the actual limit:

Conjecture 1. *For any connected graph G, $c(G^-) < 2c(G^+)$.*

It is true that $c(G^+) = 1 \rightarrow c(G^-) = 1$. Indeed, on a totally looped copwin graph, the cop never follows a loop in optimal play, as it would allow the robber to repeat the position by doing the same thing. To our knowledge, all other cases remain open.

1.3 Quadratic capture time

It was shown in [33] that for a totally looped copwin graph with n vertices, $n \geq 7$, the capture time in optimal play is at most $n-4$. For partially looped graphs, the boundary is quadratic instead of linear. We will provide an example of a sequence of partially looped grids where the capture time increases quadratically in the number of vertices.

Proposition 4. *Given a copwin graph with n vertices, the capture time in optimal play is at most $n(n-1)$.*

Proof. The number of positions of the cop and the robber is $n(n-1)$, and an optimal winning strategy for the cop does not allow repeats of the same position. $\square$

Remark 3. *The same reasoning can be applied to p cops. Also, if a graph G displays some symmetry, we can be more precise. Indeed, the group of automorphisms $Aut(G)$ acts on the set of positions of the cop(s) and robber. Since optimal winning strategies for the cop(s) do not allow repeats of the same position up to automorphism, the capture time is bounded by the number of orbits of this action.*

Example of a partially looped grid: Consider the $2 \times n$ grid, with $n \geq 5$. Its vertices shall be represented by the elements of $\{1, 2\} \times \{1, \dots, n\}$. After adding a loop on vertices $(1, 1)$ and $(1, n)$, the resulting graph will be denoted $\mathcal{G}_n$. We will show that the optimal strategy on $\mathcal{G}_n$, which is copwin, has two unusual properties: the capture time is quadratic (proposition 6), and the cop is required to move away from the robber several times in a row (corollary 3).

While these properties will seem evident to anyone who briefly examines the strategy of the cop described in proposition 7, they are surprisingly difficult to prove. Showing that this strategy is winning is easy, however, showing that it is optimal is not. The key point is to prove that the cop needs to stay on row 1 (corollary 2). Our attempts at proving it elegantly have failed. However, there is a brute force method which can be used to prove that a strategy is optimal: after computing the capture times obtained from applying this strategy for all possible starting position, it suffices to show that in any position, either the robber gets caught immediately, or the capture time decreses by 1 after each player makes their best move. While this is a simple matter of calculation, all the properties of the game can then be deduced from these capture times.

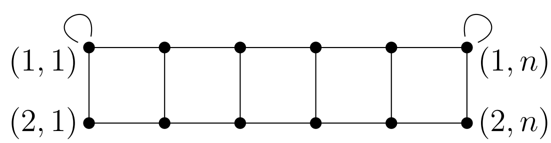


Figure 1.3: Partially looped $2 \times n$ grid

For any two vertices of $\mathcal{G}_n$, (x, a) and (y, b), we define $ct((x, a), (y, b))$ as follows:
$ct((x, a), (y, b)) = 0$ if $(x, a) = (y, b)$.
$ct((x, a), (y, b)) = 1$ if (x, a) and (y, b) are neighbours.
$ct((1, a), (1, b)) = n\frac{b-a+1}{2} - a$ if $a < b - 1$ and $a + b$ is odd.
$ct((1, a), (1, b)) = n\frac{a+b}{2} + a - 1$ if $a < b$, $a + b$ is even and $a + b \leq n$.
$ct((1, a), (1, b)) = n\frac{2n+2-a-b}{2} - a$ if $a < b$, $a + b$ is even and $a + b > n$.
$ct((1, a), (2, b)) = n\frac{b-a}{2} - a$ if $a < b$ and $a + b$ is even.
$ct((1, a), (2, b)) = n\frac{a+b-1}{2} + a - 1$ if $a < b$, $a + b$ is odd and $a + b \leq n + 1$.
$ct((1, a), (2, b)) = n\frac{2n+3-a-b}{2} - a$ if $a < b$, $a + b$ is odd and $a + b > n + 1$.
$ct((2, a), (1, b)) = n\frac{b-a+2}{2} - a + 1$ if $a < b$ and $a + b$ is even.
$ct((2, a), (1, b)) = n\frac{a+b+1}{2} + a$ if $a < b$, $a + b$ is odd and $a + b < n$.

$ct((2,a),(1,b)) = n\frac{n+1}{2} - a + 1$ if $a < b$, $a+b$ is odd and $a+b = n$.
$ct((2,a),(1,b)) = \frac{n^2}{2} + a$ if $a < b$, $a+b$ is odd and $a+b = n+1$.
$ct((2,a),(1,b)) = n\frac{2n+3-a-b}{2} - a + 1$ if $a < b$, $a+b$ is odd and $a+b > n+1$.
$ct((2,a),(2,b)) = n\frac{b-a+1}{2} - a + 1$ if $a < b-1$ and $a+b$ is odd.
$ct((2,a),(2,b)) = n\frac{a+b}{2} + a$ if $a < b-1$, $a+b$ is even and $a+b \leq n$.
$ct((2,a),(2,b)) = n\frac{2n+2-a-b}{2} - a + 1$ if $a < b-1$, $a+b$ is even and $a+b > n$.
$ct((x,a),(y,b)) = ct((x,n+1-a),(y,n+1-b))$.

Lemma 5. *If (x,a) and (y,b) are neither equal nor adjacent, then:*

$$ct((x,a),(y,b)) = 1 + \min_{(x',a')\in N((x,a))} \max_{(y',b')\in N((y,b))} ct((x',a'),(y',b')),$$

where ct is the function defined above.

The proof of lemma 5 consists of a simple case by case verification of the formula. The detailed calculations will not be fully included here, as they are both trivial and very lengthy. A couple of cases will be detailed in the appendix (section 1.6) in order to give the reader an idea of what the full version might look like. Yet, the idea that the problem can be solved in this way is more interesting than the verification itself. While this is not fully satisfying, my projects for future works include two possible alternatives: either finding a more elegant way to prove that the the cop needs to stay on the first row, or creating a machine-aided proof using an algorithm which verifies the formula in all cases.

Proposition 5. *If the cop starts in (x,a) and the robber starts in (y,b), then $ct((x,a),(y,b))$ is the capture time.*

Proof. The capture time can be defined by induction using the fact that if the cop makes her best move, then the robber makes his best move, the capture time decreases by one. This property is described by the minmax formula in lemma 5. Thus ct is the capture time. $\square$

Corollary 1. *The partially looped grid $\mathcal{G}_n$ is copwin.*

Proof. The capture time ct is finite. Hence, $\mathcal{G}_n$ is copwin. $\square$

Corollary 2. *Any optimal strategy for the cop requires her to start and stay on row 1, except for the very last move whence she may catch the robber on row 2.*

Proof. For all $a < b$, $ct((1,a),(1,b)) < ct((2,a),(2,b))$ and $ct((1,a),(2,b)) < ct((2,a),(1,b))$. Hence, when the cop moves to the second row, the robber can increase the capture time by making a vertical movement. $\square$

Proposition 6. *The capture time for $\mathcal{G}_n$ is $\left\lfloor \frac{n^2}{2} \right\rfloor$.*

Proof. The capture time for $\mathcal{G}_n$ is given by the following formula:

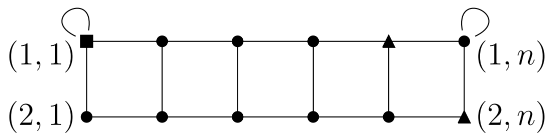
$$ct(\mathcal{G}_n) = \min_{(x,a)} \max_{(y,b)} ct((x,a),(y,b)).$$

Using the symmetry of the graph and corollary 2, we may consider only the cases where the cop starts in $(1,a)$ with $a \leq \frac{n}{2}$. If n is even, the formulas for ct give $ct((1,a),(y,b)) \leq$

$\frac{n^2}{2} + a - 1$, and this boundary is reached for $(y, b) = (1, n - a)$ or $(2, n + 1 - a)$. If n is odd, then $ct((1, a), (y, b)) \leq n\frac{n+1}{2} - a$, which is reached in $(1, n + 1 - a)$ and $(2, n + 2 - a)$.

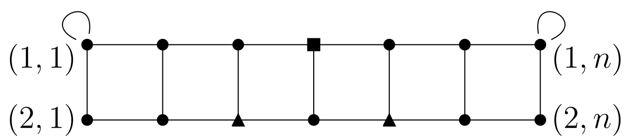
Hence, if n is even, the cop should start in the corner $(1, 1)$, and the robber in $(1, n - 1)$ or $(2, n)$, for a capture time of $\frac{n^2}{2}$ (see fig. 1.4). And if n is odd, the cop should start in the middle in $(1, \frac{n+1}{2})$ and the robber in $(2, \frac{n-1}{2})$ or $(2, \frac{n+3}{2})$, for a capture time of $\frac{n^2-1}{2}$ (see fig. 1.5).

□



■ optimal starting point for the cop
▲ optimal starting point for the robber

Figure 1.4: Optimal starting points for n even



■ optimal starting point for the cop
▲ optimal starting point for the robber

Figure 1.5: Optimal starting points for n odd

Proposition 7. *The optimal strategy for the cop on $\mathcal{G}_n$ is to go back and forth between the loops, following the loop once each time, until she can move to catch the robber. Assuming that the cop and robber both start in their optimal starting positions, the cop's first move should be to follow the loop if n is even, or to move on the first row towards the robber if n is odd.*

Proof. Corollary 2 states that in optimal play, the cop stays on the first row. Also, in optimal play, the cop can never move along the same edge twice in a row, as it would allow the robber to repeat the position by doing the same. This only leaves the cop a single possibility at each turn after the first. On the first move, if n is even, she can either follow the loop or move to $(1, 2)$. If she moves to $(1, 2)$, the robber can move to $(2, n - 1)$. Since $ct((1, 2), (2, n - 1)) = \frac{n^2}{2} + 1$, which is greater than the capture time for $\mathcal{G}_n$, this is not optimal for the cop. If n is odd and the robber starts in $(2, \frac{n+3}{2})$, the cop may go either to $(1, \frac{n-1}{2})$ or $(\frac{n+3}{2})$. In the former case, the robber may then go to $(1, \frac{n+3}{2})$, and

$ct((1, \frac{n-1}{2}), (1, \frac{n+3}{2})) = n\frac{n+1}{2} - \frac{n-1}{2} = \frac{n^2+1}{2}$. Hence, going to $(1, \frac{n-1}{2})$ is not optimal for the cop. □

Remark 4. *The cop's optimal strategy is almost independent of how the robber plays. Only her first and last moves are influenced by the robber. In fact, if we assume that the cop has limited visibility, meaning that she can only see the robber if he is in her neighbourhood, the graph $\mathcal{G}_n$ remains copwin. If the cop with limited visibility is informed of the robber's starting point, then the capture time remains $\left\lfloor \frac{n^2}{2} \right\rfloor$.*

Proposition 8. *If the cop is using her optimal strategy, an optimal strategy for the robber is to go back and forth along the path $(1,1)$, $(2,1)$, $(2,2)$, ..., $(2,n)$, $(1,n)$, following the loop once each time. If n is even, the robber will go to $(1,n)$ on his first move, then follow the loop. If n is odd, the robber will go to $(2, \frac{n+1}{2})$ on his first move.*

Proof. Since the cop's optimal strategy is known and independent of how the robberplays, we can easily verify that this strategy allows the robber to get caught in exactly $\left\lfloor \frac{n^2}{2} \right\rfloor$ moves. Since that is the maximum capture time, this strategy is optimal. □

Remark 5. *When it is the cop's turn to play, an even distance between the cop and the robber is favourable to the robber, while an odd distance is favourable to the cop. The reason why both players go back and forth between the looped vertices is that they are fighting to set the parity in their favour. However, the cop is able to follow the shortest path between the two loops, whereas the robber is forced to go via the second row in order to dodge her. So every time the cop reaches the next loop, the distance between the cop and the robber decreases by two.*

Corollary 3. *On the graph $\mathcal{G}_n$, in optimal play, the cop moves away from the robber $\left\lfloor \frac{n-3}{2} \right\rfloor$ times in a row.*

Proof. If n is odd, in optimal play, the cop moves away from the robber on turns 2 to $\frac{n-1}{2}$ when she reaches the loop. If n is even, the robber first crosses to the left of the cop on turn $\frac{n}{2} + 2$; she then moves away from him until she reaches the loop on the nth turn. □

Remark 6. *Non-looped graphs with the same properties as $\mathcal{G}_n$ may be obtained either by replacing the looped vertices with pairs of adjacent twin vertices, or by replacing the loops with triangles.*

Remark 7. *Since $\mathcal{G}_n$ has $2n$ vertices and a symmetry, the capture time $\left\lfloor \frac{n^2}{2} \right\rfloor$ tends to a quarter of the upper-bound given in remark 3 when $n \to +\infty$. When a loop is added to vertex $(2,1)$, the robber now goes back and forth between $(2,1)$ and $(1,n)$, so the path followed by the cop is shorter by 1 instead of 2. Thus the capture time is nearly doubled, for large values of n, by adding that loop. Since this also removes the symmetry, again, a quarter of the theoretical upper-bound is reached.*

1.4 Increasing the distance

We will now prove that on the graph G in fig. 1.6, there is a case in optimal play where the cop has to move away from the robber, thereby increasing the distance between them. Note that in fig. 1.6, although the loops were omitted for the sake of clarity, the graph G should be viewed as totally looped.

In order to determine the optimal capture time and starting position for the cop, we will apply the method developed in [17]. The original characterisation of totally looped copwin graphs developed in [47] and [49] involves dismantling the graphs by retracting dominated vertices one by one. The method in [17] consists of retracting every dominated vertex of the graph at each step (unless there are adjacent twin vertices dominating each other, in which case we have to leave one of the twins). The advantage of this method is that the capture time *ct* can be deduced from the number of steps required to dismantle the graph. Note that we will revisit this in more detail and using homotopies in section 2.4.

Grouping the vertices according to the step at which they are removed defines a partition of the graph called a *copwin partition*. The following result is Theorem 3.3 in [17]:

Proposition 9. *Let G have a copwin partition $X_1, X_2, \ldots, X_k$. Then, $ct(G) = k - 1$ if every vertex of X_k is adjacent to every vertex of X_{k-1}, or G has only one vertex. Otherwise, $ct(G) = k$.*

In order to illustrate this process, we number the vertices of G in the following way: first, the dismantlable vertices are labelled 1. Then, we label i the vertices which become dismantlable when all vertices with labels strictly smaller than i are removed (see fig. 1.7).

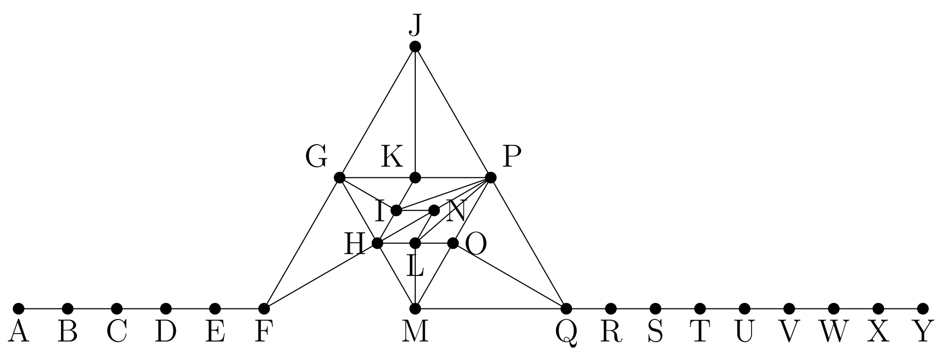


Figure 1.6: Graph G

Denoting by X_i the set of vertices labelled i, $X_1, ..., X_9$ defines a copwin partition of G. Since every vertex numbered 8 is adjacent to the only vertex numbered 9, the capture time in optimal play is 8 moves. The cop's optimal starting position is vertex Q, which has the highest label. Indeed, if the cop starts anywhere to the left (resp. right) of Q, the robber will survive longer than 8 turns by staying at Y (resp. A). Now if the robber chooses to start at J, and the cop moves either to O or P, the robber will go to G, then F and all the way to A, where he will be captured in 9 moves. Thus, unexpectedly, the optimal first move for the cop is to go to M. By doing so, she increases the distance between them

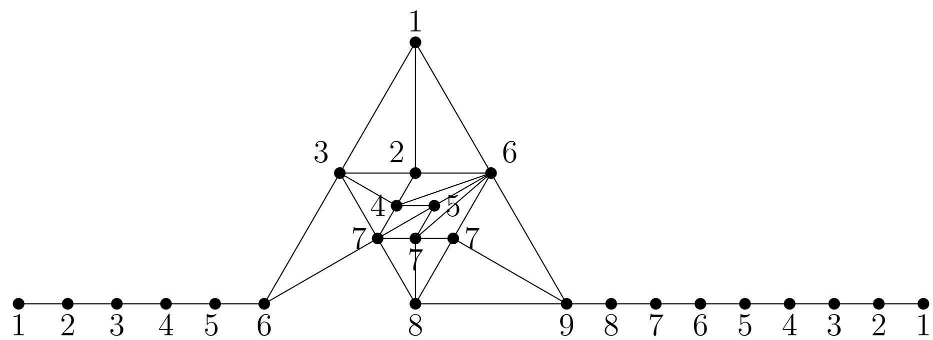


Figure 1.7: Labelling of G

from 2 to 3. The reason for this counterintuitive move is that the cop needs to cover the escape route via F before she closes in on the robber. Optimal play could proceed like this: (Q,J)→(M,P)→(O,N)→(L,I)→(N,G)→(H,K)→(I,J)→(K,J)→(J,J).

1.5 Conclusion

Our objective was to further the analysis of the effect of loops on the game of cops and robbers. This has led us to discover graphs which display various unusual properties. While passing their turn seems counterintuitive for the cops, we provided examples of graphs where preventing them from doing so almost doubles the cop number. While we showed that the cop number cannot be more than doubled with this process, we conjectured that doubling it is also impossible. This conjecture is trivial in the copwin case; yet, all other cases remain entirely open. Should this hold true, our examples would then maximise the cop number increase derived from removing the loops of a totally looped graph. We have also shown that while the capture time on totally looped graphs has a linear bound, there are very simple partially looped grids which have a quadratic capture time. On copwin graphs, an upper-bound to the capture time in optimal play is given by the number of different pairs of positions of the cop and the robber, up to isomorphism. In the examples that we give, the capture time is only a quarter of that upper-bound, so it remains to see how it can be tightened. These same examples of grids also require the cop to move away from the robber several times in a row. We have also found an example of a totally looped graph where the cop needs to move away from the robber, but making it happen several times in a row remains to be done and seems far more difficult than in the partially looped or non-looped cases. All these examples demonstrate that the case of partially looped graphs is a lot more complex than the totally looped case. The most important challenge remains to find a characterisation of partially looped copwin graphs. Understanding these unusual behaviours may eventually lead to such a characterisation.

1.6 Appendix

In this section, we verify that ct satisfies the formula which characterises the capture time in two cases. The cases where $x+y+a+b$ is odd are all similar to the first while the rest are similar to the second. The reason for this division is the *opposition* phenomenon, well-known to chess players, which describes positions where who holds the advantage is determined by whose turn it is to move [40]. In this case, the parity of $x+y+a+b$ determines which player holds the opposition. The cases are split according to the respective rows of the cop and robber, whether the cops starts to the right or to the left of the robber, and the parity of $x+y+a+b$. Special cases must be added for when either or both start on one of the four corners of the grid.

- If $x=y=1, 1<a<b-1, b<n$ and $a+b$ is odd:
  $ct((2,a),(2,b)) = n\frac{b-a+1}{2} - a + 1$
  $ct((2,a),(1,b-1)) = n\frac{b-a+1}{2} - a + 1$
  $ct((2,a),(1,b+1)) = n\frac{b-a+3}{2} - a + 1$

  $ct((1,a-1),(2,b)) = n\frac{b-a+1}{2} - a + 1$
  $ct((1,a-1),(1,b-1)) = n\frac{b-a+1}{2} - a + 1$
  $ct((1,a-1),(1,b+1)) = n\frac{b-a+3}{2} - a + 1$

  $ct((1,a+1),(2,b)) = n\frac{b-a-1}{2} - a - 1$
  $ct((1,a+1),(1,b-1)) = n\frac{b-a-1}{2} - a - 1$
  $ct((1,a+1),(1,b+1)) = n\frac{b-a+1}{2} - a - 1$

  $\max\limits_{(y',b')\in N((1,b))} ct((2,a),(y',b')) = n\frac{b-a+3}{2} - a + 1$
  $\max\limits_{(y',b')\in N((1,b))} ct((1,a-1),(y',b')) = n\frac{b-a+3}{2} - a + 1$
  $\max\limits_{(y',b')\in N((1,b))} ct((1,a+1),(y',b')) = n\frac{b-a+1}{2} - a - 1$

  Hence $\min\limits_{(x',a')\in N((1,a))} \max\limits_{(y',b')\in N((1,b))} ct((x',a'),(y',b')) = n\frac{b-a+1}{2} - a - 1.$

- If $x = y = 1, 1 < a < b-1, b < n$ and $a+b$ is even:

$$\begin{aligned}
ct((2,a),(2,b)) &= n\frac{a+b}{2} + a \text{ if } a+b \leq n \\
&= n\frac{2n+2-a-b}{2} - a + 1 \text{ if } a+b > n \\
ct((2,a),(1,b-1)) &= n\frac{a+b}{2} + a \text{ if } a+b \leq n \\
&= n\frac{n+1}{2} - a + 1 \text{ if } a+b = n+1 \\
&= \frac{n^2}{2} + a \text{ if } a+b = n+2 \\
&= n\frac{2n+4-a-b}{2} - a + 1 \text{ if } a+b > n+2 \\
ct((2,a),(1,b+1)) &= n\frac{a+b+2}{2} + a \text{ if } a+b \leq n-2 \\
&= n\frac{n+1}{2} - a + 1 \text{ if } a+b = n-1 \\
&= \frac{n^2}{2} + a \text{ if } a+b = n \\
&= n\frac{2n+2-a-b}{2} - a + 1 \text{ if } a+b > n
\end{aligned}$$

$$\begin{aligned}
ct((1,a-1),(2,b)) &= n\frac{a+b-2}{2} + a - 2 \text{ if } a+b \leq n+2 \\
&= n\frac{2n+4-a-b}{2} - a + 1 \text{ if } a+b > n+2 \\
ct((1,a-1),(1,b-1)) &= n\frac{a+b-2}{2} + a - 2 \text{ if } a+b \leq n+2 \\
&= n\frac{2n+4-a-b}{2} - a + 1 \text{ if } a+b > n+2 \\
ct((1,a-1),(1,b+1)) &= n\frac{a+b}{2} + a - 2 \text{ if } a+b \leq n \\
&= n\frac{2n+2-a-b}{2} - a + 1 \text{ if } a+b > n
\end{aligned}$$

$$
\begin{aligned}
ct((1,a+1),(2,b)) &= n\frac{a+b}{2}+a \text{ if } a+b\le n\\
&= n\frac{2n+2-a-b}{2}-a-1 \text{ if } a+b>n\\
ct((1,a+1),(1,b-1)) &= n\frac{a+b}{2}+a \text{ if } a+b\le n\\
&= n\frac{2n+2-a-b}{2}-a-1 \text{ if } a+b>n\\
ct((1,a+1),(1,b+1)) &= n\frac{a+b+2}{2}+a \text{ if } a+b\le n-2\\
&= n\frac{2n-a-b}{2}-a-1 \text{ if } a+b>n-2
\end{aligned}
$$

$$
\begin{aligned}
\max_{(y',b')\in N((1,b))} ct((2,a),(y',b')) &= n\frac{a+b+2}{2}+a \text{ if } a+b\le n-2\\
&= n\frac{n+1}{2}-a+1 \text{ if } a+b=n-1 \text{ or } n+1\\
&= \frac{n^2}{2}+a \text{ if } a+b=n \text{ or } n+2\\
&= n\frac{2n+4-a-b}{2}-a+1 \text{ if } a+b>n+2
\end{aligned}
$$

$$
\begin{aligned}
\max_{(y',b')\in N((1,b))} ct((1,a-1),(y',b')) &= n\frac{a+b}{2}+a-2 \text{ if } a+b\le n\\
&= n\frac{n+1}{2}-a+1 \text{ if } a+b=n+1\\
&= \frac{n^2}{2}+a-2 \text{ if } a+b=n+2\\
&= n\frac{2n+4-a-b}{2}-a+1 \text{ if } a+b>n+2
\end{aligned}
$$

$$
\begin{aligned}
\max_{(y',b')\in N((1,b))} ct((1,a+1),(y',b')) &= n\frac{a+b+2}{2}+a \text{ if } a+b\leq n-2\\
&= n\frac{n+1}{2}-a-1 \text{ if } a+b=n-1\\
&= \frac{n^2}{2}+a \text{ if } a+b=n\\
&= n\frac{2n+2-a-b}{2}-a-1 \text{ if } a+b>n
\end{aligned}
$$

Hence $\min_{(x',a')\in N((1,a))} \max_{(y',b')\in N((1,b))} ct((x',a'),(y',b')) = n\frac{a+b}{2}+a-2$ if $a+b\leq n$ or $n\frac{2n+2-a-b}{2}-a-1$ if $a+b>n$.

Chapter 2

Cops, Robbers and Algebraic Topology

2.1 Introduction

Perhaps the most fascinating aspect of the game of cops and robbers is that it links three very distinct areas of mathematics, ranging from the most applied to the most fundamental: game theory, graph theory and algebraic topology. What truly distinguishes the game of cops and robbers from countless other games on graphs is that it has strong algebraic properties, showing that this topic is of deep theoretical interest rather than just a simple mathematical curiosity. Already in Quilliot's thesis [49], the use of homotopies to characterise copwin graphs was established. Unfortunately, while knowledge of both the game of cops and robbers and homotopy theory of graphs has been vastly expanded since then, the combination of the two has made little progress. Quite possibly, the main obstacle is the difficulty in bridging the gap between fundamental and applied mathematics. Mathematicians who study cops and robbers mostly belong to the community of discrete mathematics, and while most are aware of the underlying notions of homotopy theory, few are attracted to this aspect of the game. Another issue is that homotopy theory of graphs is not standardised and many different notions of homotopy can be found in the literature (e.g. s-homotopy in [11], $\times$-homotopy in [22], homotopy of of digraphs in [34],...).

Nevertheless, the underlying topological idea is very simple: a cycle of length 4 or greater which is not subdivided into triangles forms a sort of hole in a graph, and the robber can evade a single cop by running around the hole. Homology groups formalise this notion of hole, and extend it to n-dimensional holes. A necessary condition for a graph to be copwin is that it should have no holes of any dimension. While less visual, homotopies give a much more precise condition as they correspond to the dismantling of the graph.

The objective of this chapter is to introduce the tools of algebraic topology in a way that is hopefully easy to understand for graph theorists who may not already be familiar with them. We will show how some well-known properties of copwin graphs can be derived from those tools. In particular, we will show how the capture time of a copwin graph can be computed using homotopies. We will also demonstrate how homotopies can be used to compare the cop numbers of graphs as well as their capture times for any number of cops.

We will start by introducing an extension of the game where instead of trying to catch the robber the usual way, the cops start on a second graph and chase his image via a graph homomorphism. This defines an extension of the game of cops and robbers which naturally extends the notions of cop number and capture time to graph homomorphisms. While the idea of having the cops chase the robber's image by a retraction is commonly used, studying the generalisation of the game will give us an interesting perspective on the initial game.

Let us recall that given a graph G, the *cop number* $c(G)$ is the minimum number of cops required to catch the robber, and the graph G is said to be *copwin* if $c(G) = 1$. We define the *capture time* $ct_k(G)$ as the minimal number of turns required for k cops to catch the robber. Note that $ct_k(G) = +\infty$ if $k < c(G)$.

In this chapter, we only consider simple graphs with loops on every vertex. The loops serve a dual purpose: they allow graph homomorphisms to send adjacent vertices to the same vertex and they allow the cops and the robber to pass their turn whenever they wish.

2.2 Cops, robbers and homomorphisms

Let $f \in Hom(X, Y)$. We consider the game where k cops are moving on Y while the robber moves on X, and the cops win if one of them is positioned on the image by f of the robber's position. We say that f is *k-copwin* if k cops have a winning strategy. The *cop number* of f, $c(f)$, is the smallest integer k such that f is k-copwin. The *capture time* $ct_k(f)$ is the number of turns required for k cops to win in optimal play, given that the robber tries to escape for as long as possible.

Remark 8. *This game is an extension of the original game, with the latter corresponding to the case $f = Id_X$. Hence, $c(X) = c(Id_X)$ and $\forall k$, $ct_k(X) = ct_k(Id_X)$.*

Proposition 10. *If $f \in Hom(X, Y)$ and $g \in Hom(Y, Z)$, then $c(g \circ f) \leq \min\{c(f), c(g)\}$ and $\forall k$, $ct_k(g \circ f) \leq \min\{ct_k(f), ct_k(g)\}$.*

Proof. If k cops on Y can catch the image by f of a robber in X in n turns, then on Z, they can apply the image by g of this strategy to catch the image of the robber by $g \circ f$ in the same number of turns. Similarly, if k cops on Z can catch the image by g of a robber in Y in n turns, they can apply the same strategy to catch the image by g of the image of a robber in X by f in the same number of turns. □

Corollary 4. *If $f \in Hom(X, Y)$, then $c(f) \leq \min\{c(X), c(Y)\}$ and $\forall k$, $ct_k(f) \leq \min\{ct_k(X), ct_k(Y)\}$.*

Proof. This result follows by taking first $f = Id_X$, then $g = Id_Y$, in proposition 10 and using remark 8. □

Remark 9. *It was shown in [42] and [50] that if X is a graph of order n, then $c(X) = O(n2^{-(1+o(1))\sqrt{log_2 n}})$. It follows from corollary 4 that this bound also applies to graph homomorphisms: $c(f) = O(n2^{-(1+o(1))\sqrt{log_2 n}})$, where $n = |range(f)|$. Similarly, it was shown in [2] that if X is planar, then $c(X) \leq 3$. It follows that if either X or Y is planar, then $c(f) \leq 3$.*

Remark 10. *The cop number is almost functorial, but not quite. Indeed, let $\mathcal{C}$ be the category with $ob(\mathcal{C}) = \mathbb{N}^*$, $hom(\mathcal{C}) = (\mathbb{N}^*)^3$, $(l, m, n) \in hom(\mathcal{C})$ has source l and target m, (n, n, n) is the identity of n and $(m, n, q) \circ (l, m, p) = (l, n, \min(p, q))$. For any graph X and any graph homomorphism $f : X \to Y$, we can define $F(X) = c(X)$ and $F(f) = (c(X), c(Y), c(f))$. If we had an equality instead of $c(g \circ f) \leq \min\{c(f), c(g)\}$, F would be a functor from the category of graphs into $\mathcal{C}$.*

The following examples show that we do not always have $c(g \circ f) = \min\{c(f), c(g)\}$. Example 2 shows that it can be false even when f and g are surjective.

Example 1. *Let C be the Hamming graph $H(3, 2)$. Let $p : C \to C$, $(a, b, c) \mapsto (a, b, 0)$ and $q : C \to C$, $(a, b, c) \mapsto (a, 0, c)$. We have $c(p) = c(q) = 2$, yet $c(q \circ p) = 1$.*

Example 2. *Given $n > m \geq 2$, let X be an $n \times n$ grid, Y an $m \times m$ torus grid and Z a cycle of length m. Let $p : X \to Y$, $(a, b) \mapsto (a \mod m, b \mod m)$ and $q : Y \to Z$, $(a, b) \mapsto a$. The homomorphisms p and q are surjective, yet we have $c(p) = c(q) = 2$ and $c(q \circ p) = 1$.*

Proposition 11. *If $f \in Hom(X, Y)$ is an isomorphism, then $c(f) = c(X)$ and $\forall k$, $ct_k(f) = ct_k(X)$. In particular, $c(X) = c(Id_X)$ and $\forall k$, $ct_k(X) = ct_k(Id_X)$.*

Proof. It follows from corollary 4 that $c(f) \leq c(X)$ and $ct_k(f) \leq ct_k(X)$. Applying proposition 10 with $g = f^{-1}$ gives the reverse inequalities. □

Proposition 12. *If $f \in Hom(X, Y)$ is constant, then $c(f) = 1$ and $\forall k$, $ct_k(f) = 0$.*

Proof. It suffices for a cop to start on the single image point of f. □

2.3 Cops, robbers and homotopies

The notion of homotopy is not standard in graph theory as it is in general topology. We use the definition given in [52]. Given $f, g \in Hom(X, Y)$, a *homotopy* between f and g of length n is an $F \in Hom(X \times P_{n+1}, Y)$ such that $F(x, 0) = f(x)$ and $F(x, n) = g(x)$. The homomorphisms f and g are then said to be *homotopic*, and we write $f \sim g$. An alternative description of a homotopy between f and g is a finite sequence $f_i \in Hom(X, Y), 0 \leq i \leq n$, such that $f_0 = f$, $f_n = g$ and if $v \in N[u]$, then $f_{i+1}(v) \in N[f_i(u)]$.

If $f \sim g$, the *homotopic distance* $d(f, g)$ is the smallest integer n for which there exists a homotopy between f and g of length n.

Proposition 13. *The homotopic distance defines a distance on each homotopy class of $Hom(X, Y)$.*

Proof. If $f \sim g$, $d(f, g) = 0$ if and only if $f = g$. Also, given a homotopy between f and g of length m and a homotopy between g and h of length n, we obtain a homotopy between f and h of length $m + n$ by concatenating the two finite sequences. Hence, $d(f, h) \leq d(f, g) + d(g, h)$. □

Remark 11. *The distance d may be extended to all of $Hom(X, Y)$ by setting $d(f, g) = +\infty$ when f and g are not homotopic, if we allow distances to be infinite.*

Theorem 1. *Let $f, g \in Hom(X, Y)$. If $f \sim g$, then $c(f) = c(g)$ and $\forall k, |ct_k(f) - ct_k(g)| \leq d(f, g)$.*

Proof. Let $F = (f_0, \dots, f_n)$ be a homotopy between f and g of length n. Assume that f is k-copwin. There is a strategy which allows k cops to catch the image of the robber by f. At this point, the robber is on a vertex r and there is a cop on $f(r)$. If the robber then moves to a vertex r' adjacent to r, since $f_0(r)$ and $f_1(r')$ are adjacent, the cop on $f(r)$ can move to $f_1(r')$. By induction, the cop who caught the robber's image by f can catch his image by f_i i turns later. Hence, $g = f_n$ is k-copwin and $ct_k(g) \leq ct_k(f) + n$. □

We have just shown that the cop number of a graph homomorphism is a homotopic invariant, while its capture time is a quasi-invariant. In the following section, we will use homotopy equivalences to evaluate and compare the usual cop numbers and capture times of graphs, which remains our main objective.

2.4 Cops, robbers and homotopy equivalences

Recall that given a graph $X = (V, E)$, a vertex u is *dominated* by vertex v if $N[u] \subset N[v]$. We define a *dominated set of vertices* as a subset $W \subset V$ such that $\forall w \in W$, $\exists v \in V \setminus W$: w is dominated by v. The set of all dominated vertices is not always a dominated set, because twin vertices may dominate each other. The practical way of identifying a maximal dominated set of vertices is to find the set of all dominated vertices, then identify twin pairs or groups of twins within that set and finally, for each group of twins, unless the vertices of that group are all dominated by another vertex, remove any one of the twins from the set.

The following definitions are analogous to standard definitions in topology: topological spaces and continuous functions are replaced with simple graphs and graph homomorphisms.

A *null-homotopy* is a graph homomorphism which is homotopic to a constant homomorphism. Two graphs X and Y are *homotopy equivalent* if there are graph homomorphisms $f : X \to Y$ and $g : Y \to X$ such that $g \circ f \sim Id_X$ and $f \circ g \sim Id_Y$. The graphs X and Y are then said to have the same *homotopy type*, and the maps f and g are called *homotopy equivalences*. A graph X is *contractible* if it has the homotopy type of a point, which is equivalent to saying that Id_X is null-homotopic. We may now reformulate the characterisation of copwin graphs ([47, 49]) using this new vocabulary:

Theorem 2. *A finite graph is copwin if and only if it is contractible.*

Proof. If a graph X is contractible, it follows from proposition 11, theorem 1 and proposition 12 that $c(X) = c(Id_X) = c(cst) = 1$, where cst is a constant homomorphism.

Conversely, if X is copwin, then it can be reduced to a single vertex by successively applying one-point retractions (see [10] p. 32). By setting $f_0 = Id_X$ and f_i equals the composition of the first i retractions, we define a homotopy between f and a constant map. □

Remark 12. *The first part of this proof also works for infinite graphs: infinite contractible graphs are copwin. However, the example (taken from [10] p. 11) of an infinite tree formed by attaching a path of each finite length to a root, is copwin but not contractible.*

The characterisation of copwin graphs is far from new and has already been stated under various forms. The initial idea consists of dismantling the graph via an elimination ordering of the vertices (see [47]). This idea then became applying successive one-point retractions (see [10] p.31). In [25], dismantlability is characterised using homotopies. The true bonus of our approach comes in the following proposition, which also characterises the capture time in terms of homotopic distance.

Proposition 14. *If $X = (V, E)$ is a finite copwin graph, then $ct_1(X) = \min_{x \in V} d(Id_X, cst_x)$, where $cst_x \in Hom(X, X)$ sends all X to x.*

Proof. Applying theorem 1 to Id_X and cst_x gives $ct_1(X) \leq d(Id_X, cst_x), \forall x \in V$.

We prove the converse by induction on the capture time. If $ct_1(X) = 0$, then X has a single vertex x and $d(Id_X, cst_x) = 0$. Assume that $ct_1(X) = n + 1$. Let W be a maximal dominated set of vertices. Whenever the cop's strategy requires her to move to a vertex of $w \in W$ before the $(n+1)$-th move, we can replace this movement with going to a vertex in $V \setminus W$ which dominates w. This modification does not affect the cature time. After applying the first n moves of this new strategy, the cop has remained in $V \setminus W$ and the robber is either caught or trapped on a dominated vertex v. In the latter case, the maximality of W implies $v \in W$. Hence, on the graph induced by $V \setminus W$, the cop can win in n turns by using this strategy. By induction, there exists $x \in V \setminus W$ and a homotopy $(f_0, \ldots, f_n)$ between $Id_{V \setminus W}$ and cst_x. Sending each element of W to a vertex in $V \setminus W$ which dominates it defines a retraction $r : V \to V \setminus W$ which is a graph homomorphism, and $(Id_X, f_0 \circ r, \ldots, f_n \circ r)$ is a homotopy of length $n + 1$ between Id_X and cst_x.

□

Remark 13. *In the proof of proposition 14, by successively substracting maximal dominated sets of vertices W_i, we define retractions $r_i : V \setminus W_i \to V \setminus W_{i+1}$ and construct a homotopy between Id_X and cst_x via $f_i = r_i \circ \ldots \circ r_0$.*

Remark 14. *A method to compute the capture time of copwin graphs was first developed in [17] (see proposition 9, which is Theorem 3.3 in [17]). Instead of using maximal dominated sets of vertices, the authors of [17] choose to deal with twin vertices in a slightly different way: at each step of the induction, they replace the vertex set with its quotient by the relation which identifies twin vertices before removing all dominated vertices.*

Theorem 3. *If X and Y are homotopy equivalent, then $c(X) = c(Y)$. And if $f \in Hom(X, Y)$ and $g \in Hom(Y, X)$ are homotopy equivalences with $d_1 = d(g \circ f, Id_X)$ and $d_2 = d(f \circ g, Id_Y)$, then $-d_2 \leq ct_k(X) - ct_k(Y) \leq d_1$, for all k.*

Proof. If f and g are homotopy equivalences between X and Y, then $c(X) = c(Id_X) = c(g \circ f) \leq c(f) \leq c(Y)$. Similarly, $c(Y) \leq c(X)$. And $ct_k(X) = ct_k(Id_X) \leq ct_k(g \circ f) + d_1 \leq ct_k(f) + d_1 \leq ct_k(Y) + d_1$. And similarly, $ct_k(Y) \leq ct_k(X) + d_2$. □

In the end, we obtained a criterion to compare not only cop numbers of graphs, but also capture times, even in cases where the number of cops is greater than the cop number.

2.5 Cops, robbers and homology

It was shown in section 2.4 that all finite copwin graphs are homotopy equivalent; more precisely, they all have the homotopy type of a point. This means that any homotopic invariant will have the same value for all finite copwin graphs. Hence, by choosing appropriate invariants, we may identify structural properties of copwin graphs. For instance, the number of connected components of a graph is a homotopic invariant; it follows that all finite copwin graphs must be connected. The objective of this section is to derive less trivial properties of finite copwin graphs by using homology groups.

In algebraic topology, singular homology groups $H_n(X), n \in \mathbb{N}$, are used to describe "holes" in a topological space X. In this section, we define homology groups of graphs specifically tailored to our purpose by replacing topological spaces and continuous functions with graphs and graph homomorphisms. Other possible approaches, which we will not detail here, include defining those homology groups via the clique complex of the graph. The cliques of a graph form a simplicial complex, so we could also define the homology of a graph as the simplicial homology of its clique complex. Alternatively, the clique complex can be viewed as a topological space, which enables us to consider its singular homology. Since all three definitions are equivalent, all the basic properties of singular homology, starting with homotopy invariance, long exact sequence and excision (see [38] pp. 110,113), can be transferred to the homology of graphs. For our purposes, we will only need the homotopy invariance, which is why we will follow the graph theoretical approach.

Let K_n denote the complete graph with n vertices numbered 0 to $n-1$, and let K_n^i denote the i-th face of K_n obtained by removing vertex i. The face K_n^i may be identified with K_{n-1} by decrementing by 1 the indices greater than i. Given a graph X, let $C_n(X)$ be the free module generated by the graph homomorphisms $\sigma : K_{n+1} \to X$; elements of $C_n(X)$ are formal linear combinations of graph homomorphisms with coefficients in $\mathbb{Z}$. Using the usual terminology of algebraic topology: for $n > 0$, the linear maps $\partial_n : C_n(X) \to C_{n-1}(X)$, $\sigma \mapsto \sum_{i=0}^{n} (-1)^i \sigma|_{K_{n+1}^i}$ and $\partial_0 : C_0(X) \to \mathbb{Z}$, $\sum_k n_k \sigma_k \mapsto \sum n_k$ are called *boundary maps.* The elements of $Ker\ \partial_n$ are called *n-cycles* (not to be confused with cycles in the graph theoretical sense) and the elements of $Im\ \partial_{n+1}$ are called *n-boundaries.*

Elements of $Hom(K_2, X)$ can be identified with directed edges of $\widetilde{X}$, where $\widetilde{X}$ is the directed graph with the same vertex set as X and a pair of edges (u, v) and (v, u) for each edge uv in X, as well as loops on every vertex. The boundary map ∂_1 sends the directed edge (u, v) to $v - u$. It follows that elements of $C_1(X)$ correspond to the labellings of the edges of $\widetilde{X}$ with finite support and with values in $\mathbb{Z}$, and the 1-cycles correspond to circulations on $\widetilde{X}$ with finite support. The loops are the boundaries of the constant homomorphisms in $Hom(K_3, X)$. Also, the boundary of the homomorphism in $Hom(K_3, X)$ which sends vertices 0 and 2 to u and 1 to v is $(v, u) - (u, u) + (u, v)$. It follows that the 1-boundaries are the circulations of $\widetilde{X}$ generated by cyclic flows based on cycles of length 1, 2 and 3.

Lemma 6. *For all $n \in \mathbb{N}$, $\partial_n \circ \partial_{n+1} = 0$; in other words, $Im\ \partial_{n+1}$ is a subgroup of $Ker\ \partial_n$.*

Proof. For $i < j$, the term in $\partial_n \circ \partial_{n+1}\sigma$ corresponding to the restriction of σ to the i-th face of K_n^j and that of the $(j-1)$-th face of K_n^i cancel out . In this fashion, all terms cancel out two by two. ☐

Based on lemma 6, we can define the *homology groups*: for all $n \in \mathbb{N}$, $H_n(X) = Ker\ \partial_n / Im\ \partial_{n+1}$. For instance, $H_1(X)$ corresponds to the classes of circulations of $\widetilde{X}$ where two circulations are equivalent if they differ by a cycle of length 1, 2 or 3. Hence, $H_1(X)$ is generated by the cycles of X which cannot be reduced to a point by successively shifting by a triangle.

Proposition 15. *If X has a single vertex, then $\forall n \in \mathbb{N}$, $H_n(X) = 0$.*

Proof. For all n, $C_n(X) = \mathbb{Z}$. If n is even, $Ker\ \partial_n = Im\ \partial_{n+1} = 0$ and if n is odd, $Ker\ \partial_n = Im\ \partial_{n+1} = \mathbb{Z}$. ☐

Proposition 16. *If X and Y are homotopy equivalent, then $\forall n \in \mathbb{N}$, $H_n(X) = H_n(Y)$.*

The proof of proposition 16 was sent to the appendix (section 2.7) as it requires some preliminary results which are not useful to the rest of this chapter.

Corollary 5. *If X is a finite copwin graph, then $\forall n \in \mathbb{N}$, $H_n(X) = 0$.*

Proof. This follows immediately from proposition 15 and proposition 16. ☐

The example shown in fig. 2.1 shows that the converse is false: $\forall n \in \mathbb{N}$, $H_n(G) = 0$ and $c(G) = 2$. It can be clearly seen that filling in the triangles of G leaves no holes, yet G is not dismantlable.

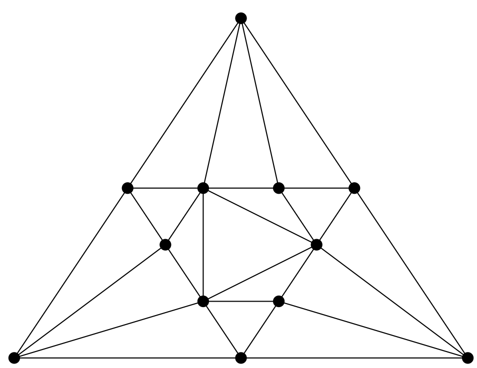


Figure 2.1: Graph G

Corollary 6. *If X is a finite copwin graph, the group of circulations of $\widetilde{X}$ is generated by cyclic flows based on cycles of lengths 1, 2 and 3.*

Proof. Since $H_1(X) = 0$, every 1-cycle is a 1-boundary. ☐

Remark 15. *This result gives a more detailed description of the structure of copwin graphs than the Triangle Lemma, Lemma 4 in [32], which states that in a copwin graph, every edge either disconnects the graph or belongs to a triangle. Indeed, the Triangle Lemma says that in every cycle of a copwin graph, every edge belongs to a triangle, whereas corollary 6 says that any cycle in a copwin graph can be reduced to a single point by successively shifting by a triangle.*

Corollary 7. *If X is a copwin plane graph, every internal face is contained inside a triangle.*

Proof. Consider a non-triangular internal face F of a copwin graph X. Since X is copwin, the perimeter of F can be reduced to a point by succesively shifting by a triangle. At some point along the process, there is a cycle which contains F that is shifted to a cycle which does not contain F. These two cycles differ by a triangle which contains F. $\square$

2.6 Conclusion and future works

Extending the notions of cop number and capture time from graphs to graph homomorphisms gives us powerful algebraic tools to study the game of cops and robbers on graphs. Homotopies between graph homomorphisms play an essential role in the study of cop numbers and capture times. The method which uses homology groups to derive structural properties of copwin graphs could perhaps be used with other homotopic invariants in order to uncover further characteristics. The game of cops and robbers on graph homomorphisms is interesting in itself and investigating its properties should be worthwhile. It shares many common properties with the standard version.

Another variant involves having both the cops and the robber moving in X, with the cops trying to have one of their images catch that of the robber. Unfortunately, the homotopy invariance does not work in that case. This variant and others involving homomorphisms should be added to the long list of interesting variants of the game of cops and robbers worth investigating.

2.7 Appendix: homotopy invariance

The following method is based on the standard proof of the homotopy invariance of singular homology (see [38] p. 120). Although the calculations seem perfectly identical, the objects used are different as topological spaces and continuous functions have been replaced with graphs and graph homomorphisms, respectively. This apparent redundancy could be circumvented by using the alternative approaches mentioned in the introduction of section 2.5, however, the method we use here has the advantage of using only graph theory without resorting to general topology.

Proposition 17. *A graph homomorphism $f \in Hom(X,Y)$ canonically induces group homomorphisms $f_\sharp : C_n(X) \to C_n(Y)$ and $f_* : H_n(X) \to H_n(Y)$, for all $n \geq 0$.*

Proof. We define $f_\sharp$ as follows: $f_\sharp : \sigma \mapsto f \circ \sigma$. Since the operations of composing with f and taking the restriction to a face commute, the diagram in fig. 2.2 is commutative:

We may describe this property using more concise notations: $f_\sharp \partial = \partial f_\sharp$. From this, we deduce that $f_\sharp$ sends n-cycles to n-cycles and n-boundaries to n-boundaries. So we define f_* as the group homomorphism which maps the equivalence class of σ to the equivalence class of $f_\sharp(\sigma)$. $\square$

Lemma 7. *i) If $f \in Hom(X,Y)$ and $g \in Hom(Y,Z)$, then $(gf)_* = g_* f_*$*
ii) $(Id_X)_* = Id_{H_n(X)}$

$$\begin{array}{ccc} C_n(X) & \xrightarrow{f_\sharp} & C_n(Y) \\ \downarrow \partial_n & & \downarrow \partial_n \\ C_{n-1}(X) & \xrightarrow{f_\sharp} & C_{n-1}(Y) \end{array}$$

Figure 2.2: Commutative Diagram

Proof. i) Follows from associativity of composition. ii) Follows immediately from the definition. □

Theorem 4. *Given $f, g \in Hom(X, Y)$, if $f \sim g$, then $f_* = g_*$.*

Proof. Let us first consider the case where $d(f, g) = 1$. In the prism $K_{n+1} \times P_2$, let us denote by $v_0, \ldots, v_n$ and $w_0, \ldots, w_n$ the vertices of $K_{n+1} \times \{0\}$ and $K_{n+1} \times \{1\}$ respectively. Let $F : X \times P_2 \to Y$ be a homotopy of length 2 between f and g. We define $\sigma \times Id : K_{n+1} \times P_2 \to X \times P_2$, $(u, v) \mapsto (\sigma(u), v)$. Using this, we now define the *prism operator* $P : C_n(X) \to C_{n+1}(Y)$ by the following formula:

$$P(\sigma) = \sum_{i=0}^{n} (-1)^i F \circ (\sigma \times Id)|_{[v_0,\ldots,v_i,w_i,\ldots,w_n]}$$

where $[v_0, \ldots, v_i, w_i, \ldots, w_n]$ denotes the subgraph of the prism induced by the enclosed vertices. We will show that $\partial P + P\partial = g_\sharp - f_\sharp$. In the following calculations, when restricting σ to the j-th face, we will use the notation $\hat{v}_j$ to indicate that vertex v_j is skipped. Let $\sigma : K_{n+1} \to X$.

$$\begin{aligned} \partial P(\sigma) = & \sum_{j \le i} (-1)^i (-1)^j F \circ (\sigma \times Id)|_{[v_0,\ldots,\hat{v}_j,\ldots,v_i,w_i,\ldots,w_n]} \\ & + \sum_{j \ge i} (-1)^i (-1)^{j+1} F \circ (\sigma \times Id)|_{[v_0,\ldots,v_i,w_i,\ldots,\hat{w}_j,\ldots,w_n]} \end{aligned}$$

The terms with $i = j$ cancel out, except for $F \circ (\sigma \times Id)|_{[\hat{v}_0,w_0,\ldots,w_n]}$ and $-F \circ (\sigma \times Id)|_{[v_0,\ldots,v_n,\hat{w}_n]}$, which are $g\sharp(\sigma)$ and $f\sharp(\sigma)$ respectively. The terms with $i \neq j$ are exactly $-P\partial(\sigma)$ since

$$\begin{aligned} P\partial(\sigma) = & \sum_{i<j} (-1)^i (-1)^j F \circ (\sigma \times Id)|_{[v_0,\ldots,v_i,w_i,\ldots,\hat{w}_j,\ldots,w_n]} \\ & + \sum_{i>j} (-1)^{i-1} (-1)^j F \circ (\sigma \times Id)|_{[v_0,\ldots,\hat{v}_j,\ldots,v_i,w_i,\ldots,w_n]}. \end{aligned}$$

Using $\partial P + P\partial = g_\sharp - f_\sharp$, if $\alpha \in C_n(X)$ is an n-cycle, then $g_\sharp(\alpha) - f_\sharp(\alpha)$ is an n-boundary. So $g_\sharp(\alpha)$ and $f_\sharp(\alpha)$ are in the same homology class, whence $f_* = g_*$.

Now if $d(f,g) = n$, there is a homotopy $F = (f_0, \ldots, f_n)$, where $f_0 = f$ and $f_n = g$. Since $d(f_i, f_{i+1}) = 1$, we have $(f_i)_* = (f_{i+1})_*$, for all i. Hence $f_* = g_*$. $\square$

We can now prove this result given in section 2.5:

Proposition 16: If X and Y are homotopy equivalent, then $\forall n \in \mathbb{N}$, $H_n(X) = H_n(Y)$.

Proof. If $f : X \to Y$ and $g : Y \to X$ are homotopy equivalences, it follows from theorem 4 that $(gf)_* = (Id_X)_*$ and $(fg)_* = (Id_Y)_*$. Using lemma 7, we deduce that f_* and g_* are inverse group isomorphisms between $H_n(X)$ and $H_n(Y)$. $\square$

Chapter 3

Firefighting on Trees[1]

3.1 Introduction and definitions

3.1.1 Context

The firefighting problem - FIREFIGHTER - was formally introduced by B. Hartnell in 1995 ([36], cited in [27]) as a variant of the game of cops and robbers. Since then, it has raised the interest of many researchers. While this game started as a very simple model for fire spread and containment problems for wildfires, it can also represent any kind of threat able to spread sequentially in a network (diseases, viruses, rumours, flood . . .).

Like the game of cops and robbers, it is a deterministic discrete-time game defined on a graph. However, unlike the game of cops and robbers, it is a one-player game. In the beginning, a fire breaks out on a vertex and at each step, if not blocked, the fire spreads to all adjacent vertices. In order to contain the fire, the player is given a number f_i of firefighters at each turn i and can use them to protect vertices which are neither burning nor already protected. The game terminates when the fire cannot spread any further. In the case of finite graphs, the aim is to save as many vertices as possible; while in the infinite case, the player wins if the game finishes, which means that the fire is contained.

This problem and its variants give rise to a generous literature; the reader is referred to [27] for a broad presentation of the main research directions. A significant amount of theoretical work deals with its complexity and approximability behaviour in various classes of graphs [7, 15, 26, 30] and its parametrised complexity (e.g. [6, 15]). In particular, when one firefighter is available at each turn it is known to be polynomially solvable in some classes of graphs, which include graphs of maximum degree 3 if the fire breaks out on a vertex of degree at most 2 [26], interval graphs, permutation graphs and split graphs [30]. However it is known to be very hard, even in some restrictive cases. In particular, the case of trees was revealed to be very rich and a lot of research focuses on it. The problem, with the same

[1] This chapter is based on a collaborative work between Bertrand Jouve (IMT Toulouse, France), Pierre Coupechoux (LAAS-CNRS, Toulouse, France), my supervisor Marc Demange (RMIT University, Melbourne, Australia) and myself (RMIT University, Melbourne, Australia). The results were presented by Pierre at the International Symposium on Combinatorial Optimization, ISCO 2018. A 12-page version was published in conference proceedings [19]. A full version, submitted to the journal Theoretical Computer Science, is currently in press [20].

number of firefighters at each turn, is NP-hard on finite trees of maximum degree 3 [26], as well as in even more restricted cases [7]; the reader is also referred to [16] for further complexity results. Regarding approximation results on trees, a greedy strategy was first shown to be a $\frac{1}{2}$-approximation algorithm [37] if a fixed number of firefighters is available at each turn. For a single firefighter, a $(1-\frac{1}{e})$-approximation algorithm is proposed in [15] for the problem in trees. This ratio was improved in [39] for ternary trees and, very recently, a polynomial time approximation scheme was obtained in trees [1]. This essentially closes the question of approximating the firefighter problem in trees with one firefighter and motivates considering some generalisations. The problem is hard to approximate within $n^{1-\varepsilon}$ on general graphs and with a single firefighter [3].

Most papers on this subject deal with a constant firefighter sequence. In fact, the problem was originally defined with one firefighter per turn. The case of infinite grids is of particular interest and has led to the model being extended by varying the available resources per turn. The change was motivated by the fact that a fire of any size on an infinite grid can be contained with two firefighters per turn but not with one [28, 53]. In order to refine these results, M.-E. Messinger started considering periodic firefighter sequences [45] while more general sequences are considered in [24]. A related research direction investigates integer linear programming models for the problem, especially on trees [1, 35, 44]. This line of research makes very natural a *relaxed* version where the amount of firefighters available at each turn is any non-negative real number and the amount allocated to vertices lies between 0 and 1. A vertex with a protection less than 1 is *partially protected* and its unprotected part can burn partially and transmit only its fraction of fire to the adjacent vertices. Thus, the f_i may take any non-negative value. This defines a variant game called FRACTIONAL FIREFIGHTER which was introduced in [28].

3.1.2 Our contribution

The main thread of this chapter is the focus on general firefighter sequences, which raises three specific research questions. We address these questions when a single fire spreads throughout a rooted tree.

First, we introduce an online version of both FIREFIGHTER and FRACTIONAL FIREFIGHTER where the sequence of firefighters is revealed over time (online) while the graph (a tree in our case) is known from the start. To our knowledge, this is the first attempt at analysing online firefighter problems. Although our motivation is mainly theoretical, this paradigm is particularly natural in emergency management where one has to make quick decisions despite lack of information. Any progress in this direction tells us how lack of information impacts the quality of the solution. Note that a version of the game introduced in [9] also models a lack of information. In that version, rather than the firefighting resources, the missing information is where the fire will spread. Also, randomised analyses are proposed to maximise the expected number of saved vertices while we use worst case analyses expressed in terms of competitive ratios.

A second question, the *separating problem*, deals specifically with infinite trees. Separating two given firefighter sequences means finding an infinite tree on which the fire can be contained with one sequence but not the other.

The third question deals with criteria for the fire to be contained based on the asymptotic behaviours of the firefighter sequence and the size of the levels in the tree. Unlike the first two questions, it has already been investigated in some articles (e.g., [23, 41]) for Firefighter with firefighter sequences of the form (λ^n).

The chapter is organised as follows: in section 3.2 we define formally Firefighter and Fractional Firefighter as well as their online versions. Section 3.3 deals with competitive analysis when the fire spreads in a finite tree and the firefighter sequence is revealed online. We first generalise an analysis of a greedy algorithm known only in special cases of Firefighter to Fractional Firefighter. For the offline case, it answers an open question proposed in [27, 35]. Then we propose improved competitive algorithms for online Firefighter with a small total number of firefighters while establishing that the greedy approach is optimal in the general case. The last two sections (section 3.4 and section 3.5) both deal with the infinite case. Section 3.4 deals with our second question. Considering the class of spherically symmetric trees where all vertices at the same level have the same degree, we express the separation problem as a purely numerical one-player game, which we call the *targeting game*. We propose two sufficient conditions for the existence of a winning strategy. Section 3.5 deals with our third question. We establish sufficient conditions for containing the fire expressed as asymptotic comparisons of the number of available firefighters and the size of the levels in the tree. In the online case, for a particular class of trees the level size of which grows linearly, we also give a sufficient condition to contain the fire.

3.1.3 Some notations

Let T be a tree rooted in r. Given two vertices v and v', $v \lhd v'$ denotes that v is an ancestor of v' (or v' is a descendant of v) and $v \unlhd v'$ denotes that either $v = v'$ or $v \lhd v'$. For any vertex v, let $T[v]$ denote the sub-tree induced by v and its descendants. Let T_i denote the i-th level of T rooted in r, where $\{r\} = T_0$. For a finite tree T rooted in r, the height $h(T)$ is the maximum length of a path from r to a leaf. If $i > h(T)$, we have $T_i = \emptyset$. The *weight* $w(v)$ of a vertex v is the number of vertices of $T[v]$. When no ambiguity may occur, we will simply write $w_v = w(v)$.

We denote by $B(T)$ the tree obtained from T by contracting all vertices from levels 0 and 1 into a new root vertex r_B: for all $u_1 \in T_1$ and $u_2 \in T_2$, every edge ru_1 is contracted and every edge $u_1u_2 \in E(T)$ gives rise to an edge $r_Bu_2 \in E(B(T))$. For $k \leq h(T)$, $B^k(T)$ will denote the k^{th} iteration of B applied to T: all vertices from levels 0 to k are contracted into a single vertex denoted by r_{B^k} which becomes the new root.

3.2 Problems and preliminary results

3.2.1 Firefighter and Fractional Firefighter

An instance of Fractional Firefighter is defined by a triple $(G, r, (f_i))$, where $G = (V(G), E(G))$ is a graph, $r \in V(G)$ is the vertex where the fire breaks out and $(f_i, i \geq 1)$ is the non-negative *firefighter sequence*. Note that the game could be extended by allowing negative values for f_i, however, we will exclude pyromaniac firefighters, with one exception in section 3.4.2 for the purpose of simplifying a proof. Let S_i denote the cumulative amount of firefighters received: $S_i = \sum_{j=1}^{i} f_j$.

At turn $i = 0$, the game is in its initial state where r is burning and all other vertices are unprotected, and $i \geq 1$ corresponds to the different rounds of the game. At each turn $i \geq 1$ and for every vertex v, the player decides which amount $p_i(v)$ of protection to add to v. Throughout the game, for every vertex v the part of v which is burning at turn $i \geq 0$ is denoted by $b_i(v)$, with $b_0(r) = 1$ and $b_0(v) = 0$ for all $v \neq r$. Similarly the cumulative protection received by vertex v is $p_i^c(v)$ with $p_0^c(v) = 0$ for all v. Both $(b_i(v), i \geq 0)$ and $(p_i^c(v), i \geq 0)$ are non-decreasing sequences with $b_i(v)+p_i^c(v) \leq 1$ for all i and v. At each turn i, the player's choice of $p_i(v)$ is subject to the constraints $p_i(v) \geq 0$, $b_{i-1}(v)+p_{i-1}^c(v)+p_i(v) \leq 1$ and $\sum_{v \in V(G)} p_i(v) \leq f_i$. The new protection of v is $p_i^c(v) = p_{i-1}^c(v) + p_i(v)$. The fire then spreads following the rule

$$b_i(v) = \max\{\max_{v' \in N(v)} b_{i-1}(v') - p_i^c(v), b_{i-1}(v)\}, \tag{3.1}$$

where $N(v)$ denotes the open neighbourhood of v. The game finishes when the fire stops spreading (i.e. $b_i(v) = b_{i-1}(v)$ for all v).

We show that this game always terminates on a finite graph G. Let $L_r(G)$ denote the maximum length of an induced path in G with extremity r, we have:

Proposition 18 ([20]). *The maximum number of turns before a game of* Firefighter *or* Fractional Firefighter *on a finite graph G will terminate is $L_r(G)$.*

Proof. First, we show by induction on i that:

> For all $i \geq 1$, for all vertex v, if $b_i(v) > b_{i-1}(v)$, then there is an induced path $P_{i,v} = (u_0 = r, u_1, \ldots, u_i = v)$ of length i such that $b_j(u_j)$ is non-increasing along the path.

For $i = 1$ and $v \in V(G)$, if $b_1(v) > b_0(v)$ then v is a neighbour of r and $b_0(r) = 1 \geq b_1(v)$. The path $P_{1,v} = (r, v)$ is of length 1.

Suppose the property holds at turn i and that $b_{i+1}(v) > b_i(v)$ for some v. Necessarily, v receives the additional amount of fire from a neighbour w:

$$b_{i+1}(v) = b_i(w) - p_{i+1}^c(v) > b_i(v).$$

We necessarily have $b_i(w) > b_{i-1}(w)$ since otherwise $b_i(w) = b_{i-1}(w)$ and we would have the following contradiction:

$$b_i(v) \geq b_{i-1}(w) - p_i^c(v) = b_i(w) - p_i^c(v) \geq b_i(w) - p_{i+1}^c(v) > b_i(v).$$

Applying to w the induction hypothesis, there is an induced path $P_{i,w} = (u_0 = r, u_1, \ldots, u_i = w)$ such that $(b_j(u_j))$ is non-increasing. Since, $b_{i+1}(v) = b_i(w) - p_{i+1}^c(v)$, we have $b_{i+1}(v) \leq b_i(w)$.

Also, for all $j < i$, u_j and v are not adjacent since otherwise, we would have the following contradiction:

$$b_i(v) \geq b_{j+1}(v) \geq b_j(u_j) - p^c_{j+1}(v) \geq b_i(w) - p^c_{i+1}(v) > b_i(v).$$

It follows that the path $P_{i+1,v}$ obtained by adding the edge wv to $P_{i,w}$ is an induced path which satisfies the required property.
Thus, if $b_{i+1}(v) > b_i(v)$ for some v, we have $i + 1 \leq L_r(G)$. Consequently, the fire can no longer spread at turn $L_r(G) + 1$.

Conversely, let P be an induced path with extremity r of length $L_r(G)$. If $(f_i) = (|G| - L_r(G) - 1, 0, 0, 0 \cdots)$ and if the complement of P is protected during the first turn, the game will terminate in exactly $L_r(G)$ turns. □

Example 3. *On a perfect binary tree (B_n, r) of height n, given one firefighter per turn, the length of* (Fractional) Firefighter *is exactly $n = L_r(B_n)$, whatever the player's strategy (this will be an immediate consequence of proposition 21).*

The standard Firefighter problem is similar to Fractional Firefighter, but with the additional constraint that the $p_i(v)$ are all binary variables. It follows that $p^c_i(v)$ and $b_i(v)$ are also binary. In this case, the firefighter sequence must have integral values.

3.2.2 Simplification for trees

Given an instance $(T, r, (f_i))$, where T is a tree, T will be considered rooted in r. In order to remove trivial cases, we will exclude algorithms which place at turn i more protection on a vertex v than the part of v that would burn if no protection were placed starting from turn i. If T is finite, proposition 18 implies that the game will end in at most $h(T)$ turns. We consider that it has exactly $h(T)$ turns, eventually with empty turns where no firefighters are allocated towards the end of the game.

Solutions on trees have a very specific structure. Indeed, it immediately follows from eq. (3.1) that, when playing on a tree, at each turn i, the amounts of fire $b_i(v)$ are non-increasing along any path from the root, which means that the fire will only spread outwards from the root. Also, for every vertex v in T_j for some j, the amount of fire $b_j(v)$ can no longer increase after turn j. Hence, no protection is placed in T_j at turn $i > j$. Note also that for any solution which allocates a positive amount of protection at turn i to a vertex $v \in T_k, k > i$, allocating the same amount of protection to the parent of v instead strictly improves the performance. Indeed, if vertex v can still burn, so can its parent. So we may consider only algorithms that play in T_i at turn i. For an optimal algorithm, this property was emphasised in [37].

This holds for both Firefighter and Fractional Firefighter on trees. For such an algorithm, $p^c_i(v) = p_i(v)$ and the values of $p_i(v)$ and $b_i(v)$ will not change after turn i for $v \in T_i$. Hence, the index i may be dropped by denoting $p(v) = p_i(v)$ and $b(v) = b_i(v)$ for $v \in T_i$. A solution p is then characterised by the values $p(v), v \in V(T)$. For any solution p, while $p(v)$ represents the amount of protection received directly, vertex v also receives protection through its ancestors, the amount of which is denoted by $P_p(v) = \sum_{v' \lhd v} p(v')$ (used in section 3.3.1). Since we only consider algorithms that play in T_i at turn i and do not place extraneous protection, for any vertex v, $p(v) + P_p(v) \leq 1$. Also, for any vertex $v \in T_i$, we have $b_{i-1}(v) = 0$ and by summing eq. (3.1) from 1 to i, we deduce that $p(v) + P_p(v) + b(v) = 1$.

Any solution p for Firefighter or Fractional Firefighter will satisfy the constraints:

$$[\mathcal{C}] \begin{cases} \sum_{v \in T_i} p(v) \leq f_i & (i) \\ \forall v, p(v) + P_p(v) \leq 1 & (ii) \end{cases}$$

In [44], a specific boolean linear model has been proposed for solving Firefighter on a tree T involving these constraints. Solving Fractional Firefighter on T corresponds to solving the relaxed version of this linear program.

3.2.3 Online version

Online optimisation [4] is a generalisation of approximation theory which represents situations where the information is revealed over time and one needs to make irrevocable decisions. We now introduce online versions of Firefighter and Fractional Firefighter on trees. The graph and starting point of the fire are known from the start, but the firefighter sequence $(f_i)_{i \geq 1}$ is revealed over time. This set-up can be seen as a game between the online player (or algorithm) and a malicious *adversary*. At each turn i, the adversary reveals f_i and then the player chooses where to allocate protection. We refer to the usual case, where $(f_i)_{i \geq 1}$ is known in advance, as *offline*.

Let us consider an online algorithm **OA** for one of the two problems and let us play the game on a finite tree T until the fire stops spreading. The value $\lambda_{\mathbf{OA}}$ achieved by the algorithm, defined as the amount of saved vertices, is measured against the best value performed by an algorithm which knows in advance the sequence (f_i). In the present case, it is simply the optimal value of the offline instance, referred to as the *offline optimal value*, denoted by β_I when considering the online Firefighter (I stands for "Integral") and β_F for the online Fractional Firefighter. We will call **Bob** such an algorithm, able to see the future and guaranteeing the value β_I or β_F for online Firefighter and Fractional Firefighter.

Algorithm **OA** is said to be γ-competitive, $\gamma \in]0, 1]$, for the online Firefighter (resp. Fractional Firefighter) if for every instance, $\frac{\lambda_{\mathbf{OA}}}{\beta_I} \geq \gamma$ (resp. $\frac{\lambda_{\mathbf{OA}}}{\beta_F} \geq \gamma$); γ is also called a *competitive ratio* guaranteed by **OA**. An online algorithm will be called *optimal* if it guarantees the best possible competitive ratio.

The arguments given in section 3.2.2, to justify considering only algorithms that play in T_i at turn i, are still valid for online algorithms; thus we will only consider such online algorithms. Let us start with a reduction:

Proposition 19 ([20])**.** *We can reduce online* (Fractional) Firefighter *on trees to instances where* $f_1 > 0$.

Proof. If $f_i = 0$ for all i such that $1 \leq i \leq k$, then the instance $(T, r, (f_i))$ is equivalent to the instance $(B^k(T), r_{B^k}, (f_{i+k}))$. □

In the infinite case we do not define competitive ratios, but only ask whether the fire can be contained by an online algorithm. Sections 3.3.1 and 3.3.2 deal with the finite case while section 3.5.3 deals with a class of infinite trees.

3.3 Online firefighting on finite trees

3.3.1 Competitive analysis of a greedy algorithm

Greedy algorithms are usually very good candidates for online algorithms, sometimes the only known approach. Mainly two different greedy algorithms have been considered in the literature for Firefighter on a tree [27] and they are both possible online strategies in our set-up. The *degree greedy* strategy prioritises saving vertices of large degree; it has been shown in [7] that it cannot guarantee any approximation ratio on trees, even for a constant firefighter sequence. A second greedy algorithm, introduced in [37] for an integral sequence (f_i), maximises at each turn the total weight of the newly protected vertices. We generalise it to any firefighter sequence for both the integral and the fractional problems. Let **GR** denote the greedy algorithm that selects at each turn i an optimal solution of the linear program $\mathcal{P}_i$ with variables $x(v), v \in T_i$ and constraints $[\mathcal{C}]$:

$$\mathcal{P}_i : \begin{cases} \max \sum_{v \in T_i} x(v)w(v) & \\ \sum_{v \in T_i} x(v) \leq f_i & (i) \\ \forall v \in T_i, x(v) + P_x(v) \leq 1 & (ii) \end{cases}$$

An optimal solution of $\mathcal{P}_i$ is obtained by ordering vertices $\{v_1, \ldots, v_{|T_i|}\}$ of level i by non-increasing weight and taking them one by one in this order and greedily assigning to vertex v_j the value $x(v_j) = \min(f_i - \sum_{k<j} x(v_k), 1 - P_x(v_j))$. Note that **GR** is valid for both Firefighter and Fractional Firefighter.

It was shown in [37] that the greedy algorithm on trees gives a $\frac{1}{2}$-approximation of the restriction of Firefighter when a single firefighter is available at each turn. They claim that this approximation ratio remains valid for a fixed number $D \in \mathbb{N}$ of firefighters at each turn. We extend this result to any firefighter sequence $(f_i)_{i \geq 1}$, integral or not. Since **GR** is an online algorithm, the performance can also be seen as a competitive ratio for the online version.

Theorem 5 ([20])**.** *The greedy algorithm* **GR** *is $\frac{1}{2}$-competitive for both online* Firefighter *and* Fractional Firefighter *on finite trees.*

Proof. Let us first consider the fractional case with an online instance $(T, r, (f_i))$ of Fractional Firefighter on a tree.

Let $x(v)$ and $y(v)$ be the amounts of firefighters placed on vertex v by **GR** and **Bob**, respectively. We have $\lambda_{\mathbf{GR}} = \sum_v x(v)w(v)$ and $\beta_F = \sum_v y(v)w(v)$.

Recall that $P_x(v) = \sum_{v' \lhd v} x(v')$ and $P_y(v) = \sum_{v' \lhd v} y(v')$. We split $y(v)$ into two non-negative quantities, $y(v) = g(v) + h(v)$, where $g(v)$ is the part of $y(v)$ already protected by **GR** through the ancestors of v, while $h(v)$ is the part of $y(v)$ which, when added on top of $P_y(v)$, exceeds $P_x(v)$. So, if $P_y(v) < P_x(v) < y + P_y(v)$, we have $h(v) = y(v) - P_x(v)$. The general formula is:
$g(v) = \min\{y(v), \max\{0, P_x(v) - P_y(v)\}\}$ and $h(v) = \max\{0, y(v) + \min\{0, P_y(v) - P_x(v)\}\}$.

We now claim that $\forall v' \in T$, $\sum_{v \unlhd v'} g(v) \leq P_x(v')$ and prove it by induction. Since $g(r) = 0$, it holds for the root r. Assuming that the inequality holds for a vertex v', let v''

be a child of v'. If $P_x(v'') - P_y(v'') \geq 0$, then we directly have:

$$\sum_{v \trianglelefteq v''} g(v) = \sum_{v \triangleleft v''} g(v) + g(v'') \leq \sum_{v \triangleleft v''} y(v) + (P_x(v'') - P_y(v'')) = P_x(v'').$$

Else, $g(v'') = 0$ and using $\sum_{v \trianglelefteq v'} g(v) \leq P_x(v')$ and $P_x(v'') \geq P_x(v')$, the inequality holds for v''; which completes the proof of the claim. Thus:

$$\sum_{v'} \sum_{v \trianglelefteq v'} g(v) \leq \sum_{v'} P_x(v') = \sum_{v'} \sum_{v \triangleleft v'} x(v) \leq \sum_{v'} \sum_{v \trianglelefteq v'} x(v).$$

Since $w(v) = \sum_{v \trianglelefteq v'} 1$, by changing the order of summation on both sides, we obtain:

$$\sum_{v} g(v)w(v) \leq \sum_{v} x(v)w(v) = \lambda_{\mathbf{GR}}. \tag{3.2}$$

Let us now consider the coefficients $h(v)$. We claim that the coefficients $h(v)$ with $v \in T_i$ satisfy the constraints (i) and (ii) of $\mathcal{P}_i$: indeed for (i), we have $h(v) \leq y(v)$ and y satisfies constraint (i). For (ii) note that $h(v) + P_x(v) = \max\{P_x(v), y(v) + \min\{P_x(v), P_y(v)\}\} \leq \max\{P_x(v), y(v) + P_y(v)\} \leq 1$.

Hence, $\forall i, \sum_{v \in T_i} h(v)w(v) \leq \sum_{v \in T_i} x(v)w(v)$ and therefore:

$$\sum_{v \in T} h(v)w(v) \leq \sum_{v \in T} x(v)w(v) = \lambda_{\mathbf{GR}}. \tag{3.3}$$

Finally, since $g(v) + h(v) = y(v)$, we conclude from eqs. (3.2) and (3.3) that $\beta_F \leq 2\lambda_{\mathbf{GR}}$. Hence the Greedy algorithm is $\frac{1}{2}$-competitive for the online Fractional Firefighter problem. Since the greedy algorithm gives an integral solution if (f_i) has integral values and since $\beta_F \geq \beta_I$, it is also $\frac{1}{2}$-competitive for the Firefighter problem. This concludes the proof of theorem 5.

□

Conjecture 2.3 in [35] (which is also Conjecture 3.5 in [27]) claims that there is a constant ρ such that the optimal value of Fractional Firefighter is at most ρ times the optimal value of Firefighter. It was supported by extensive experimental tests [35], but finding such a constant and proving the ratio is one of the open problems proposed in [27] (Problem 7). Theorem 5 can be expressed as $\lambda_{\mathbf{GR}} \leq \beta_I \leq \beta_F \leq 2\lambda_{\mathbf{GR}}$, which shows that $\rho = 2$ is such a constant:

Corollary 8 ([20])**.** *In* Fractional Firefighter*, the amount of vertices saved is at most twice the maximum number of vertices saved in* Firefighter*.*

3.3.2 Improved competitive algorithm for Firefighter

In this section, we investigate possible improvements for online strategies for Firefighter on finite trees. Let $\varphi = \frac{1+\sqrt{5}}{2}$ denote the golden ratio, satisfying $\varphi^2 = \varphi + 1$.

For any integer $k \geq 2$, we denote by $\alpha_{I,k}$ the best possible competitive ratio for online Firefighter on finite trees if at most k firefighters are available in the entire game. We have:

$$\alpha_{I,k} = \inf_{T \in \mathcal{T}} \max_{\mathbf{OA} \in \mathcal{A}_L} \min_{(f_i) \in \mathbb{N}^{\mathbb{N}^*}, \sum_i f_i \leq k} \frac{\lambda_{\mathbf{OA}}}{\beta_I},$$

where $\mathcal{T}$ denotes the set of finite rooted trees and $\mathcal{A}_L$ the set of online algorithms for Firefighter on finite trees.

Note that in the definition of $\alpha_{I,k}$, $\lambda_{\mathbf{OA}}$ and β_I depend on T. Also, the maximum and the minimum are well defined since on a finite tree T, the set of possible ratios is finite. An online algorithm, choosing for any fixed T a strategy which achieves this maximum, will be $\alpha_{I,k}$-competitive for instances with at most k firefighters. Such an algorithm is optimal for these instances.

The sequence $(\alpha_{I,k})$ is non-increasing. We define $\alpha_I = \lim_{k \to \infty} \alpha_{I,k}$; again, the index I stands for *Integral* and refers to the Firefighter problem.

Remark 16. *The limit α_I is the greatest competitive ratio that can be reached on any tree. Indeed, given a finite tree T, it suffices to consider the instances with at most $|V(T)|$ firefighters.*

In this section, we give an online algorithm for instances of Firefighter on a finite tree that is optimal (i.e., $\alpha_{I,2}$-competitive) if at most two firefighters are presented. Based on proposition 19, we may assume $f_1 \neq 0$. If $f_1 = 2$, one firefighter will be called the first and the other one the second. An online instance is then characterised by when the second firefighter is presented. It can be never if only one firefighter is presented or at the first turn if $f_1 = 2$. Note that this later case is trivial since an online algorithm can make the same decision as **Bob** by assigning both firefighters to two unburnt vertices of maximum weights. Our algorithm works also in this case and will make this optimal decision.

Lemma 8 ([20])**.** *Let a and b be two vertices of maximum weights in T_1. If $\sum_i f_i \leq 2$, there is an optimal offline algorithm for* Firefighter *which places the first firefighter on either a or b.*

Proof. If the first firefighter is placed on $v \in T_1 \setminus \{a, b\}$ by an optimal offline algorithm, since at most two firefighters are available, $\exists u \in \{a, b\}$, $T[u]$ burns completely. Hence, replacing v by u when assigning the first firefighter would produce another optimal solution (necessarily $w_v = w_u$). □

We suppose **Bob** has this property. However, even if $w_a > w_b$, he will not necessarily choose a; as illustrated by the graph $W_{1,10,20}$ (fig. 3.1) where if the firefighter sequence is $(1, 0, 1, 0, 0, 0 \ldots)$, then **Bob**'s needs to protect x during the first turn. Note also that, when the root is of degree at least 3, the second firefighter is not necessarily in $V(T[a]) \cup V(T[b])$.

We now consider algorithm 1 and assume that the adversary will reveal at most two firefighters. The algorithm works on an updated version $\widetilde{T}$ of the tree: if one vertex is protected, then the corresponding sub-tree is removed and all the burnt vertices are contracted into the new root $\tilde{r}$ so that the algorithm always considers vertices of level 1 in $\widetilde{T}$. Before starting the online process, the algorithm computes the weights of all vertices. The weights of the unburnt vertices do not change when updating $\widetilde{T}$. The value of $h(\widetilde{T})$, required in line 9, can be computed during the initial calculation of weights and easily updated with $\widetilde{T}$. For the sake of clarity, we do not detail all updates in the algorithm.

In this section only, for any vertex $v \in T_i$ and any $i \leq j \leq i + h(T[v])$, we denote by v_j a vertex of maximum weight w_{v_j} in $T_j \cap V(T[v])$, i.e. among the descendants of v which are in level j (or v itself if $i = j$). We also define $\bar{w}_{v_j}$ for all j via:

$$\bar{w}_{v_j} = w_{v_j} \text{ if } j \in [i; i + h(T[v])] \text{ and } 0 \text{ otherwise.}$$

Algorithm 1

Require: A finite tree T with root r - An online adversary.

1: $(\widetilde{T}, \tilde{r}) \leftarrow (T, r)$; Compute w_v, $\forall v \in V(\widetilde{T})$
2: $First\_Firefighter \leftarrow TRUE$;
3: {Start of the online process}
4: At each turn, after the fire spreads, $\widetilde{T}$ is updated - burnt vertices are contracted to $\tilde{r}$;
5: If several firefighters are presented at the same time, we consider them one by one in the following lines;
6: **if** a new Firefighter is presented **and** $\tilde{r}$ has at least one child **then**
7:   **if** $First\_Firefighter$ **then**
8:     Let a and b denote two children of $\tilde{r}$ with maximum weight w_a, w_b and $w_a \geq w_b$ ($a = b$ if $\tilde{r}$ has only one child);
9:     **if** $\min\limits_{2 \leq i \leq 1+h(\widetilde{T})} \frac{w_a + \bar{w}_{b_i}}{w_b + \bar{w}_{a_i}} \geq \frac{1}{\varphi}$ **then**
10:       Place the first firefighter on a;
11:     **else**
12:       Place the first firefighter on b;
13:     $First\_Firefighter \leftarrow FALSE$;
14:   **else**
15:     Place the firefighter on a child v of $\tilde{r}$ of maximum weight

Theorem 6 ([20])**.** *Algorithm 1 is a $\frac{1}{\varphi}$-competitive online algorithm for online* FIREFIGHTER *with at most two firefighters available. It is optimal for this case.*

Proof. While algorithm 1 runs feasibly on any instance, we limit the analysis to the case where at most two firefighters are available. If the adversary does not present any firefighter before the turn $h(T)$, both algorithm 1 and **Bob** cannot save any vertex and, by convention, the competitive ratio is 1.

Let us suppose that at least one firefighter is presented at some turn $k \leq h(T)$; the tree still has at least one unburnt vertex. During the first $(k-1)$ turns, the instance is updated to $(B^{k-1}(T), r_{B^{k-1}}, (f_{i+k-1}))$. In the updated instance, at least one firefighter is presented during the first turn and the root has at least one child. Proposition 19 ensures that it is equivalent to the original instance.

If the root $r_{B^{k-1}}$ has only one child, line 8 gives $a = b$ and algorithm 1 selects a at line 10. In the updated instance, all vertices are saved; so the competitive ratio is equal to 1.

Else, we have $a \neq b$ with $w_a \geq w_b$ (line 8). If the adversary presents a single firefighter for the whole game, then algorithm 1 protects either a or b. Meanwhile, **Bob** will protect a, saving w_a vertices. If $w_a \geq \varphi w_b$, then we have:

$$\forall i, 2 \leq i \leq 1 + h(T), \frac{w_a + \bar{w}_{b_i}}{w_b + \bar{w}_{a_i}} \geq \frac{w_a}{w_b + w_a} \geq \frac{\varphi w_b}{w_b + \varphi w_b} = \frac{1}{\varphi}. \tag{3.4}$$

So algorithm 1 protects a (line 9), guaranteeing a competitive ratio of 1. Otherwise, if $w_b > \frac{1}{\varphi} w_a$, even placing the firefighter on b guarantees a ratio of at least $\frac{1}{\varphi}$.

Suppose now that the adversary presents two firefighters. We consider two cases.

Case (i): If algorithm 1 places the first firefighter on a at line 10, and if the adversary presents the second firefighter at turn $i \geq k$, then the algorithm will save $w_a + \bar{w}_{x_i}$, for some $x \in T_k \setminus \{a\}$ such that $\bar{w}_{x_i} = \max_{u \in T_k \setminus \{a\}} \bar{w}_{u_i}$. For the same instance, **Bob** will save $w_v + \bar{w}_{y_i}$ for some $v \in \{a, b\}$ and $y \in T_k \setminus \{v\}$. If the two values are different (the optimal one is strictly better), then necessarily $v = b$ and $y = a$. In this case the criterion of line 9 ensures that the related competitive ratio is at least $\frac{1}{\varphi}$.

Case (ii): Suppose now that algorithm 1 places the first firefighter on b at line 12, and say the adversary presents the second firefighter at turn $j \geq k$. Lines 8 and 9 ensure that:

$$\exists i, 2 \leq i \leq 1 + h(T), \frac{w_a + \bar{w}_{b_i}}{w_b + \bar{w}_{a_i}} < \frac{1}{\varphi}. \tag{3.5}$$

Hence, we have $w_a < \varphi w_b$, since in the opposite case, eq. (3.4) would hold. Algorithm 1 now saves $w_b + \bar{w}_{x_j}$, for $x \in T_k \setminus \{b\}$ such that $\bar{w}_{x_j} = \max_{u \in T_k \setminus \{b\}} \bar{w}_{u_j}$. Meanwhile, **Bob** selects $v \in \{a, b\}$ and, if it exists, y_j for some $y \in T_k \setminus \{v\}$, for a total of $w_v + \bar{w}_{y_j}$ vertices saved. If $y \neq b$, then $\bar{w}_{y_j} \leq \bar{w}_{x_j}$, by definition of x, and thus:

$$\frac{w_b + \bar{w}_{x_j}}{w_a + \bar{w}_{y_j}} \geq \frac{w_b + \bar{w}_{x_j}}{w_a + \bar{w}_{x_j}} \geq \frac{w_b}{w_a} > \frac{1}{\varphi}. \tag{3.6}$$

Finally, if $y = b$, then $v = a$ and the competitive ratio to evaluate is $\frac{w_b + \bar{w}_{x_j}}{w_a + \bar{w}_{b_j}}$. We claim that the following holds:

$$\frac{w_a + \bar{w}_{b_i}}{w_b + \bar{w}_{a_i}} \times \frac{w_b + \bar{w}_{x_j}}{w_a + \bar{w}_{b_j}} \geq \frac{1}{\varphi^2}. \tag{3.7}$$

If $i \geq j$, then $\bar{w}_{a_i} \leq \bar{w}_{a_j}$ and since $a \neq b$, $\bar{w}_{a_j} \leq \bar{w}_{x_j}$. Hence: $\frac{w_b + \bar{w}_{x_j}}{w_b + \bar{w}_{a_i}} \geq \frac{w_b + \bar{w}_{x_j}}{w_b + \bar{w}_{a_j}} \geq 1$ and therefore:

$$\frac{w_a + \bar{w}_{b_i}}{w_b + \bar{w}_{a_i}} \times \frac{w_b + \bar{w}_{x_j}}{w_a + \bar{w}_{b_j}} \geq \frac{w_a + \bar{w}_{b_i}}{w_a + \bar{w}_{b_j}} \geq \frac{w_a}{w_a + w_b}.$$

Now, if $i < j$, we get: $\frac{w_a + \bar{w}_{b_i}}{w_a + \bar{w}_{b_j}} \geq 1$ and therefore:

$$\frac{w_a + \bar{w}_{b_i}}{w_b + \bar{w}_{a_i}} \times \frac{w_b + \bar{w}_{x_j}}{w_a + \bar{w}_{b_j}} \geq \frac{w_b + \bar{w}_{x_j}}{w_b + \bar{w}_{a_i}} \geq \frac{w_b}{w_a + w_b}.$$

In both cases, since $\frac{w_a}{w_a + w_b} \geq \frac{w_b}{w_a + w_b} \geq \frac{1}{1+\varphi} = \frac{1}{\varphi^2}$, we obtain eq. (3.7). Now, eqs. (3.5) and (3.7) imply that in case (ii), when $y = b$, we also have $\frac{w_b + \bar{w}_{x_j}}{w_a + \bar{w}_{b_j}} \geq \frac{1}{\varphi}$. Together with eq. (3.6), this concludes case (ii) and shows that algorithm 1 is $\frac{1}{\varphi}$-competitive. □

Even though complexity analyses are not usually proposed for online algorithms, it is worth noting that line 9 only requires the weights of vertices in $V(T[a]) \cup V(T[b])$ and the maximum weight per level in $T[a]$ and $T[b]$. Hence, algorithm 1 requires $O(|V(T[a])| + |V(T[b])|)$ to choose the position of the first firefighter and $O(|V(T)|)$ altogether.

We conclude this section with a hardness result justifying that the greedy algorithm **GR** is optimal and that algorithm 1 is optimal if at most two firefighters are available. These hardness results will all be derived from the graphs $W_{k,l,m}$ (fig. 3.1).

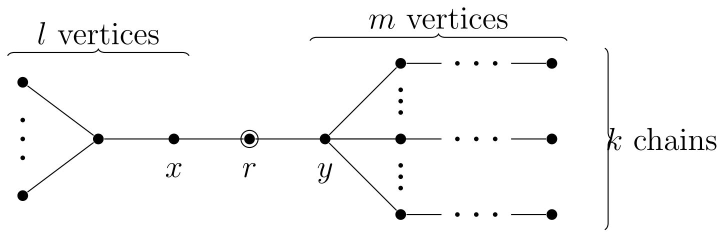


Figure 3.1: Graph $W_{k,l,m}$

Proposition 20 ([20])**.** *For all* $k \geq 2, \frac{1}{2} \leq \alpha_{I,k} \leq \frac{1}{\varphi}$, *more precisely:*
(i) $\alpha_I = \frac{1}{2}$, *which means that the greedy algorithm is optimal for* FIREFIGHTER *in finite trees;*
(ii) $\alpha_{I,2} = \frac{1}{\varphi}$, *which means that algorithm 1 is optimal if at most two firefighters are available;*
(iii) $\alpha_{I,4} < \frac{1}{\varphi}$.

Proof. Theorem 5 shows that $\alpha_I \geq \frac{1}{2}$. Given integers l, m, k such that $k | m-1$, we define the graph $W_{k,l,m}$ as shown in fig. 3.1. We will assume that $m > k^2$.

(i) Let us consider an online algorithm for $W_{k,l,m}$. As established in section 3.2.3 we can assume that the online algorithm plays in T_i at turn i. If $f_1 = 1$, the algorithm will protect either x or y. If x is selected and the firefighter sequence is $(1, 1, 0, 0, 0, \ldots)$, our online algorithm protects the branch of x and one of the k chains, while the optimal offline algorithm protects y and the star. Its performance is then $\frac{l+\frac{m-1}{k}}{l+m-1}$. If, however, y is protected instead during the first turn and if the firefighter sequence is $(1, 0, 1, 1, 1, \ldots)$, the online algorithm protects the branch of y and one vertex of the star whilst the optimal algorithm protects the branch of x as well as the k chains, minus $\frac{k(k+1)}{2}$ vertices. If $l = m - 1 = k^4$, for large values of k, the online algorithm which protects x is more performant and its competitive ratio is $\frac{1+\frac{1}{k}}{2}$. Having $k \to +\infty$ shows that $\alpha \leq \frac{1}{2}$. Since the greedy algorithm **GR** guarantees $\alpha_I \geq \frac{1}{2}$, we have $\alpha_I = \frac{1}{2}$.

(ii) Consider the graphs $W_{1,l,\lfloor \varphi l \rfloor}$. If the online algorithm protects x, the adversary selects the sequence $(1, 0, 0, 0, \ldots)$, whereas if the online algorithm protects y, $(1, 0, 1, 0, 0, 0, \ldots)$ is selected. In both cases, the performance tends to $\frac{1}{\varphi}$ when $l \to +\infty$.

(iii) If at most 4 firefighters are available, the graph $W_{4,901,1001}$ gives an example where $\frac{1}{\varphi}$ cannot be reached. Indeed, if $f_1 = 1$ and the online algorithm protects x, then the adversary will select the sequence $(1, 1, 0, 0, 0, \ldots)$, as in the proof of *(i)*, for a performance of $\frac{1151}{1901}$. If the online algorithm protects y, since firefighters are limited to 4, the adversary

will select $(1,0,1,1,1,0,0,0,\ldots)$, for a performance of $\frac{1002}{1645}$. This second choice is slightly better; however $\frac{1002}{1645} < \frac{1}{\varphi}$. □

I have also proved that there is a $\frac{1}{\varphi}$-competitive algorithm if three firefighters are presented (i.e., $\alpha_{I,3} = \frac{1}{\varphi}$). This algorithm is similar to algorithm 1 in that it places the first firefighter on one of the three largest branches and greedily places each of the other two on the largest branch available at the time they are presented. The proof uses the fact that the optimal online and offline algorithms both place the first firefighter on one of the three largest branches available. The possible cases branch out further as the next firefighters may or may not be placed on descendants of vertices protected by the other algorithm, just like in the case for two firefighters, but with many more variations. At the time when I proved it, I did not know that $\alpha_{I,4} = \frac{1}{\varphi}$, and I was striving to find a $\frac{1}{\varphi}$-competitive algorithm for any number of firefighters. With hindsight, the result for three firefighters is a lot less interesting since cannot lead to any generalisation. Thus, the lengthy proof has not been included here.

3.4 Separating firefighter sequences

3.4.1 Definitions

We now consider the fractional firefighter problem on infinite graphs. We say that a sequence of firefighters (f_i) is *weaker* than (f'_i) (or (f'_i) is *stronger* than (f_i)) if $\forall k, S_k \leq S'_k = \sum_{i=1}^{k} f'_i$, and we write $(f_i) \preceq (f'_i)$. If we also have $\exists k : S_k < S'_k$, (f_i) is said to be *strictly weaker* than (f'_i) and we write $(f_i) \prec (f'_i)$.

Lemma 9 ([20]). *If the fire can be contained in the instance $(G, r, (f_i))$ and if $(f_i) \preceq (f'_i)$, then the fire can also be contained in $(G, r, (f'_i))$ by an online algorithm that knows (f_i) in advance.*

Proof. Given a winning strategy in the instance $(G, r, (f_i))$, if (f'_i) firefighters are available, we contain the fire by protecting the same vertices, possibly earlier than in the initial strategy. □

However, if $(f_i) \prec (f'_i)$, for Fractional Firefighter, it is not always the case that there is an infinite graph G such that the fire can be contained in $(G, r, (f'_i))$ but not in $(G, r, (f_i))$ (see example 4). We call such a G a *separating graph* for (f_i) and (f'_i), and we say that G *separates* (f_i) and (f'_i) in N turns if the fire can be contained in N turns for (f'_i) but not for (f_i). In this section, we give sufficient conditions for the existence of a separating graph.

Example 4. *Let $f_1 = 1$, $f'_1 = 1.5$ and $\forall i \geq 2$, $f_i = f'_i = 0$. Although $(f_i) \prec (f'_i)$, no graph separates those two sequences.*

Note that for Firefighter, the problem is trivial, as shown in corollary 9.

3.4.2 Spherically symmetric trees

Given a sequence $(a_i) \in (\mathbb{N}^*)^{\mathbb{N}^*}$, the *spherically symmetric tree* $T((a_i))$ is the tree rooted in r where every vertex of level $i-1$ has a_i children [43]. Note that if $T = T((a_i))$, we have $|T_i| = \prod_{j=1}^{i} a_j$. The total amount of fire at level i is the sum of the amounts of fire on all vertices of level i.

Proposition 21 ([20]). *In the instance $(T((a_i)), r, (f_i))$, the total amount of fire that spreads to level i is max$\{0, F_i\}$, where $F_0 = 1$ and $F_i = a_i F_{i-1} - f_i$ for all i.*

Proof. At turn i, the player only protects vertices on level i. If no protection is placed, the total amount of fire is multiplied by a_i. Hence, if $f_i \geq a_i F_{i-1}$, the fire is contained; else, the total amount of fire spreading to level i is $a_i F_{i-1} - f_i$, regardless of how the protection is distributed among the vertices of level i. □

Corollary 9 ([20]). *Let (f_i) and (f'_i) be two distinct integral valued sequences. There is a spherically symmetrical tree which separates (f_i) and (f'_i).*

Proof. Let k be the first rank where $f_k \neq f'_k$. We may assume that $f_k < f'_k$. It follows from proposition 21 that in the instance $(T((f_i + 1)), r, (f_i))$, the amount of fire that spreads to each level is equal to 1. Yet, in $(T((f_i + 1)), r, (f'_i))$, the fire is contained at turn k. □

For the purpose of the following technical lemma, we define a new firefighter sequence, which may include a negative term. Given a firefighter sequence (f_i) and two non-zero integers k and ϵ, we define the firefighter sequence $(f_i^{(k,\epsilon)})$ via:

$$f_i^{(k,\epsilon)} = \begin{cases} f_k + \epsilon & \text{if } i = k \\ f_{k+1} - \epsilon & \text{if } i = k+1 \\ f_i & \text{otherwise} \end{cases}$$

Note that there is a possibility that $f_{k+1}^{(k,\epsilon)}$ might be negative; however, this does not impact the reasoning. We also define the sequence $(F_i^{(k,\epsilon)})$ via $F_0^{(k,\epsilon)} = 1$ and $F_i^{(k,\epsilon)} = a_i F_{i-1}^{(k,\epsilon)} - f_i^{(k,\epsilon)}$. It follows from proposition 21 that the amount of fire which spreads to level i in the instance $(T((a_i)), r, (f_i^{(k,\epsilon)}))$ is $\max\{0, F_i^{(k,\epsilon)}\}$.

Lemma 10 ([20]). *The spherically symmetric tree $T = T((a_i))$ separates (f_i) and $(f_i^{(k,\epsilon)})$ if and only if there is a rank N such that: $A \leq \sum_{i=k+2}^{N} \dfrac{f_i}{\prod_{j=k+2}^{i} a_j} < B$, where $A = F_{k+1}^{(k,\epsilon)}$ and $B = F_{k+1}$.*

Proof. It follows from proposition 21 that $F_n = \prod_{j=1}^{n} a_j - \sum_{i=1}^{n} f_i \prod_{j=i+1}^{n} a_j$. So, $F_n = |T_n|(1 - \sum_{i=1}^{n} \frac{f_i}{|T_i|})$. The condition for $T((a_i))$ to separate (f_i) and $(f^{(k,\epsilon)})$ can be stated as follows: there is a rank N such that $F_N^{(k,\epsilon)} \leq 0 < F_N$. Hence, there is an N such that

$$|T_N|(1 - \sum_{i=1}^{N} \frac{f_i^{(k,\epsilon)}}{|T_i|}) \leq 0 < |T_N|(1 - \sum_{i=1}^{N} \frac{f_i}{|T_i|}).$$

Therefore

$$1-\sum_{i=1}^{k+1}\frac{f_i^{(k,\epsilon)}}{|T_i|}\leq\sum_{i=k+2}^{N}\frac{f_i}{|T_i|}<1-\sum_{i=1}^{k+1}\frac{f_i}{|T_i|}.$$

And finally,

$$A\leq\sum_{i=k+2}^{N}\frac{f_i}{\prod_{j=k+2}^{i}a_j}<B,$$

with $A=F_{k+1}^{(k,\epsilon)}=|T_{k+1}|(1-\sum_{i=1}^{k+1}\frac{f_i^{(k,\epsilon)}}{|T_i|})$ and $B=F_{k+1}=|T_{k+1}|(1-\sum_{i=1}^{k+1}\frac{f_i}{|T_i|})$. □

Proposition 22 ([20]). *Given two sequences (f_i) and (f'_i) such that $(f_i)\preceq(f'_i)$, let k be the smallest integer such that $f_k\neq f'_k$ and let $\epsilon=f'_k-f_k$. The spherically symmetric tree $T=T((a_i))$ separates (f_i) and (f'_i) if and only if there is a rank N such that: $A\leq\sum_{i=k+2}^{N}\dfrac{f_i}{\prod_{j=k+2}^{i}a_j}<B$,*
where $A=F_{k+1}^{(k,\epsilon)}$ and $B=F_{k+1}$.

Proof. We have $(f_i^{(k,\epsilon)})\preceq(f'_i)$, as indeed, $(f_i^{(k,\epsilon)})$ is the weakest sequence in $\{(g_i)\in\mathbb{R}^{\mathbb{N}}|\forall i<k,\ g_i=f_i,\ g_k=f_k+\epsilon \text{ and } (f_i)\preceq(g_i)\}$. Hence, any tree separating (f_i) and $(f_i^{(k,\epsilon)})$ also separates (f_i) and (f'_i). We conclude using lemma 10. □

3.4.3 Targeting game

Given the form of the condition in proposition 22, we can view satisfying it as a special case of a purely numerical problem, which we will call the *targeting game*. The instance of the problem is given by two positive real numbers, $A<B$, and a sequence of non-negative real numbers (f_i) which represents the movements towards the target $[A,B[$. The player starts at position $u_0=0$ with an initial step size of 1. We denote by δ_i the step size at turn i, so $\delta_0=1$. At each turn $i>0$, the player chooses a positive integer a_i by which he will divide the previous step size, that is to say $\delta_i=\dfrac{\delta_{i-1}}{a_i}=\prod_{j=1}^{i}a_j^{-1}$. Then, the position of the player is updated with the rule $u_i=u_{i-1}+f_i\delta_i$. If there is an integer N such that $u_N\in[A,B[$, then the player wins with the strategy (a_i).

The targeting game can be summarised as follows: Given $0<A<B$ and a sequence (f_i), is there an N and a sequence (a_i) such that $A\leq\sum_{i=1}^{N}\dfrac{f_i}{\prod_{j=1}^{i}a_j}<B$?

We give two sufficient conditions on the data to ensure the existence of a winning strategy for the player.

Theorem 7 ([20]). *If there is an N such that $\sum_{i=1}^{N}f_i\geq A\left\lceil\dfrac{A}{B-A}\right\rceil$, then there exists a sequence (a_i) with $a_i=1,\forall i\geq 2$ such that the player wins the targeting game at turn N by selecting (a_i).*

Proof. First, note that if $m \geq \dfrac{A}{B-A}$, then $(m+1)A \leq mB$. It follows that

$$[A\left\lceil \frac{A}{B-A} \right\rceil, +\infty[\subset \bigcup_{k\in\mathbb{N}^*} [kA, kB[.$$

Hence, there is a k such that $kA \leq \sum_{i=1}^N f_i < kB$. So, $A \leq \sum_{i=1}^N \dfrac{f_i}{k} < B$. Therefore, if the player chooses $a_1 = k$ and $a_i = 1$, for $i \geq 2$, he will have reached the target at turn N. □

Theorem 8 ([20]). *If $|\{i : f_i \geq B\}| \geq \log_2\left(\dfrac{B}{B-A}\right)$, then the player wins the targeting game by choosing at each turn i the smallest positive integer a_i such that $u_i < B$.*

Proof. Consider a turn i such that $a_i > 1$. Given that the player chooses the minimum a_i, it follows that $B \leq u_{i-1} + \dfrac{\delta_{i-1}}{a_i - 1} f_i$. By definition of u_i and δ_i, we have $\delta_{i-1} f_i = a_i(u_i - u_{i-1})$, so $B \leq u_{i-1} + \dfrac{a_i}{a_i - 1}(u_i - u_{i-1})$. Then $B(a_i - 1) \leq u_{i-1}(a_i - 1) + a_i(u_i - u_{i-1})$ and

$$a_i(B - u_i) \leq B - u_{i-1}. \tag{3.8}$$

Now consider the sequence $x_i = \dfrac{B - u_i}{\delta_i}$. By dividing eq. (3.8) by δ_{i-1}, we see that $x_i \leq x_{i-1}$ when $a_i > 1$. When $a_i = 1$, we also have $x_i = x_{i-1} - f_i \leq x_{i-1}$. Thus (x_i) is non-increasing, and $\forall i, x_i \leq x_0 = B$.

At any turn i where $f_i \geq B$, we have $f_i \geq x_{i-1}$ and

$$\delta_{i-1} f_i \geq \delta_{i-1} x_{i-1} = B - u_{i-1} > u_i - u_{i-1} = \delta_i f_i.$$

So $\delta_{i-1} > \delta_i$, and $a_i > 1$. It then follows from eq. (3.8) that $B - u_i \leq \dfrac{B - u_{i-1}}{2}$.

Note also that, since (x_i) and (δ_i) are non-increasing, $(B - u_i)$ is also non-increasing. Hence, for all N, we have:

$$B - u_N \leq \frac{B - u_0}{2^{|\{i \leq N : f_i \geq B\}|}} = \frac{B}{2^{|\{i \leq N : f_i \geq B\}|}}.$$

Finally, choosing N such that $|\{i \leq N : f_i \geq B\}| \geq \log_2\left(\dfrac{B}{B-A}\right)$, we have $A \leq u_N < B$. □

Proposition 23 ([20]). *Given $(f_i) < (f'_i)$, let k be the smallest integer such that $f_k \neq f'_k$ and let $\epsilon = f'_k - f_k$. If there is an N such that $\sum_{k+2}^N f_i \geq 2\left\lceil \dfrac{2}{\epsilon} \right\rceil$ or $|\{k+2 \leq i \leq N; f_i \geq 2\}| > 1 - log_2\epsilon$, then there is a spherically symmetric tree which separates (f_i) and (f'_i) in N turns.*

Proof. For $i \le k$, we choose the smallest a_i such that $F_i > 0$; i.e. $a_i = \left\lfloor \frac{f_i}{F_{i-1}} \right\rfloor + 1$. We then choose $a_{k+1} = \max\{2, \left\lfloor \frac{f_{k+1}}{F_k} \right\rfloor + 1\}$.

Using proposition 22, it is sufficient to have a rank N such that:

$$A \le \sum_{i=k+2}^{N} \frac{f_i}{\prod_{j=k+2}^{i} a_j} < B,$$

where $A = |T_{k+1}|(1 - \sum_{i=1}^{k+1} \frac{f_i^{(k,\epsilon)}}{|T_i|})$ and $B = |T_{k+1}|(1 - \sum_{i=1}^{k+1} \frac{f_i}{|T_i|})$

It follows from the choice of a_i that for $i \le k$, $(a_i - 1)F_{i-1} - f_i \le 0$. Hence, $a_i F_{i-1} - f_i \le F_{i-1}$, and $F_i \le F_{i-1}$. If $a_{k+1} = \left\lfloor \frac{f_{k+1}}{F_k} \right\rfloor + 1$, then $F_{k+1} \le F_k$. Otherwise, if $a_{k+1} = 2$, $F_{k+1} \le 2F_k$. Finally, we have $B = F_{k+1} \le 2F_0 = 2$.

Also, $B - A = \sum_{i=1}^{k+1} (f_i^{(k,\epsilon)} - f_i) \prod_{j=i+1}^{k+1} a_j = (f_k^{(k,\epsilon)} - f_k)a_{k+1} + (f_{k+1}^{(k,\epsilon)} - f_{k+1}) = (a_{k+1} - 1)\epsilon$. Having chosen $a_{k+1} \ge 2$, we have $B - A \ge \epsilon$.

Thus, we have an N such that $\sum_{k+2}^{N} f_i \ge 2 \left\lceil \frac{2}{\epsilon} \right\rceil \ge A \left\lceil \frac{A}{B - A} \right\rceil$ or $|\{k + 2 \le i \le N | f_i \ge 2\}| > 1 - log_2\epsilon \ge \log_2 \left(\frac{B}{B - A} \right)$. The result follows by applying theorem 7 or theorem 8.

□

Remark 17. *In the case where $|\{k + 2 \le i \le N; f_i \ge 2\}| > 1 - log_2\epsilon$, the sequence (a_i) is entirely created by a greedy algorithm which selects the minimum value of a_i such that $F_i > 0$ (and $a_{k+1} \ge 2$). The value of a_i is therefore a function of F_{i-1} and f_i.*

Special credit for the targeting game should be given to Pierre Coupechoux, as it was his idea. After studying it together, we began wondering if it might possibly have applications other than the firefighter game. We first tried the field of robotics, after imagining a mechanical arm that would be aimed towards a target using several gears. However, PhD students who actually study robotics informed us that they saw no reason for the parameters of any such construct to be limited to integers. We imagined a rabbit hopping towards its burrow playing the targeting game. While the successive hops and the target interval are there, why would the jumping strength of the rabbit be divided by integers? Replacing the rabbit with a kangaroo appealed to those of us residing in Australia, but did not otherwise help. A hot air balloon dropping unusually weighted ballast could explain the integers, unfortunately, hot air balloons do not hop. In the end, it was Marc who found the correct interpretation:

In a far away land, rebellious mathematicians who tried to subvert the mathematical society are held prisoners in a penitentiary where they are made to break rocks all day long. The prison warden knows that many rebels remain on the outside and wishes to have the inmates betray their fellow rebels. He offers them a deal: one day, they are given a large number of huge rocks to break into rubble. After finishing a rock, they are allowed to betray some of their friends who will then join them in their task, though to be fair, each inmate should betray the same number of rebel friends. The warden promises them that if they manage to complete the task during dusk on the third day of the trial, they will all obtain full pardon and a reduction of their teaching load.

3.5 Firefighting sequence vs. sevel growth

3.5.1 Infinite offline instances

On infinite trees, the objective is to contain the fire. Hence, finite branches of infinite trees are irrelevant. In particular, the problem is trivial on infinite trees without infinite branches and we exclude this case. Given an infinite rooted tree T with at least one infinite branch, let us consider the union T^* of all leafless sub-trees of T with the same root, where the union $T_1 \cup T_2$ of two sub-trees of T is the sub-tree induced by $V(T_1) \cup V(T_2)$. T^* is leafless (it is obtained from T by pruning finite branches). Playing on T is equivalent to playing on T^*. Without loss of generality, we may restrict the infinite case to leafless trees. Note that if T is leafless, then $(|T_i|)$ is non-decreasing.

Intuitively, it seems that when the fire cannot be contained on an infinite tree, it means that the number of vertices per level must grow faster, in some sense, than the firefighter sequence. Following this line of reasoning, proposition 24 and theorem 9 give criteria for infinite instances to be winning based on the asymptotic behaviours of those two sequences.

Proposition 24 ([20]). *Let $(T, r, (f_i))$ be an instance of* Fractional Firefighter *where T is a tree of infinite height. If $\sum_{i=1}^{+\infty} \frac{f_i}{|T_i|} > 1$, then the instance is winning.*

Proof. The firefighter wins by spreading at each turn n the amount of protection evenly among all vertices of T_n. The amount of fire that reaches $v \in T_n$ is $\max\{0, 1 - \sum_{i \le n} \frac{f_i}{|T_i|}\}$. Hence, the fire is contained after a finite number of turns. □

Unfortunately, we need a more complex criterion to obtain a sufficient condition to win in both Firefighter and Fractional Firefighter.

Theorem 9 ([20]). *Let $(T, r, (f_i))$ be an instance of* Firefighter *or* Fractional Firefighter *where T is a leafless tree. If $S_i \to +\infty$ and $\frac{S_i}{|T_i|} \nrightarrow 0$, then the instance $(T, r, (f_i))$ is winning for the firefighter.*

The proof of theorem 9 will require the following lemma, which is probably well-known:

Lemma 11 ([20]). *If (u_n) is a positive sequence that increases towards $+\infty$, then $\sum \frac{u_n - u_{n-1}}{u_n}$ diverges.*

Proof. Let $v_n = \frac{u_n - u_{n-1}}{u_n}$. If $v_n \nrightarrow 0$, then $\sum v_n$ diverges. Let us assume that $v_n \to 0$. Since $\frac{u_{n-1}}{u_n} = 1 - v_n$, we have $ln \frac{u_0}{u_n} = ln \prod_{i=1}^{n} 1 - v_i = \sum_{i=1}^{n} ln(1 - v_i)$. Since $u_n \to +\infty$, $ln \frac{u_0}{u_n} \to -\infty$, so $\sum ln(1 - v_i) \to -\infty$, and since $v_n \to 0$, $\sum_{i=1}^{n} v_i \to +\infty$. □

We may now prove theorem 9.

Proof. Since T is leafless, $(|T_i|)$ is non-decreasing and since $\frac{S_i}{|T_i|} \nrightarrow 0$, there is a positive constant C and an increasing injection $\sigma : \mathbb{N} \to \mathbb{N}$ such that

$$\forall i, |T_{\sigma(i)}| \le C S_{\sigma(i)} < \infty.$$

Let $a : V(T) \to [0, 1]$ denote the amount of protection we will place on each vertex. In order to describe the amount of each vertex that remains unprotected at the end of each turn, we use a sequence of labellings $l_i : V(T) \to [0,1]$. Initially, all vertices are unprotected, so $l_0 = 1$. At turn i, protection is placed on vertices of T_i, so $\forall v \in V(T), l_i(v) = l_{i-1}(v) - \sum_{v' \in T_i} a(v')\mathbb{1}_{v' \trianglelefteq v}$. For any $W \subset V(T)$ and any labelling l, we define $l(W) = \sum_{v \in W} l(v)$.

For all i and $h \in \mathbb{N}^*$ with $h > i$, for all $v \in T_i$, let $w_h(v) = |\{v' \in T_h, v \triangleleft v'\}|$. Thus, $\sum_{v \in T_i} w_h(v) = |T_h|$ and for all $j < i$,

$$\sum_{v \in T_i} w_h(v) l_j(v) = l_j(T_h). \tag{3.9}$$

It follows from lemma 11 that $\sum \frac{S_{\sigma(i)} - S_{\sigma(i-1)}}{S_{\sigma(i)}}$ diverges. Hence $\prod(1 + \frac{S_{\sigma(i)} - S_{\sigma(i-1)}}{CS_{\sigma(i)}})$ also diverges. Let N be such that $\prod_{i=1}^{N}(1 + \frac{S_{\sigma(i)} - S_{\sigma(i-1)}}{CS_{\sigma(i)}}) > 2C$ and let h be such that $S_{\sigma(h)} > 2S_{\sigma(N)}$.

We consider the following strategy. At each turn i, we protect the vertices which have the most descendants in level $\sigma(h)$, i.e., $a(v), v \in T_i$, is an optimal solution of the following linear program:

$$\begin{cases} \max \sum_{v \in T_i} a(v) w_{\sigma(h)}(v) \\ a(v) \leq l_{i-1}(v) \ ; \ v \in T_i \\ \sum_{v \in T_i} a(v) \leq f_i \end{cases}$$

If $f_i \geq l_{i-1}(T_i)$ then we can protect the whole level T_i, thus $T_{\sigma(h)}$, and the fire is contained. So we assume $f_i < l_{i-1}(T_i)$ for all $i < \sigma(h)$. Then, $f_i \frac{l_{i-1}(v)}{l_{i-1}(T_i)}, v \in T_i$ is a solution of the linear program with $\sum_{v \in T_i} f_i \frac{l_{i-1}(v)}{l_{i-1}(T_i)} = f_i$. It follows, by optimality and using eq. (3.9), that:

$$\sum_{v \in T_i} a(v) w_{\sigma(h)}(v) \geq \sum_{v \in T_i} f_i \frac{l_{i-1}(v)}{l_{i-1}(T_i)} w_{\sigma(h)}(v) = \frac{f_i l_{i-1}(T_{\sigma(h)})}{l_{i-1}(T_i)}$$

Note that for $j \leq i$, $l_{j-1}(T_j) \leq |T_j| \leq |T_{\sigma(i)}|$. Hence,

$$\begin{aligned} l_{\sigma(i-1)}(T_{\sigma(h)}) - l_{\sigma(i)}(T_{\sigma(h)}) &= \sum_{j=\sigma(i-1)+1}^{\sigma(i)} \sum_{v \in T_j} a(v) w_{\sigma(h)}(v) \\ &\geq \sum_{j=\sigma(i-1)+1}^{\sigma(i)} \frac{f_j l_{j-1}(T_{\sigma(h)})}{l_{j-1}(T_j)} \\ &\geq \sum_{j=\sigma(i-1)+1}^{\sigma(i)} \frac{f_j l_{\sigma(i)}(T_{\sigma(h)})}{|T_{\sigma(i)}|} \\ &\geq \frac{S_{\sigma(i)} - S_{\sigma(i-1)}}{CS_{\sigma(i)}} l_{\sigma(i)}(T_{\sigma(h)}) \quad \text{(since } |T_{\sigma(i)}| \leq CS_{\sigma(i)}\text{)}. \end{aligned}$$

So

$$l_{\sigma(i-1)}(T_{\sigma(h)}) \geq (1 + \frac{S_{\sigma(i)} - S_{\sigma(i-1)}}{CS_{\sigma(i)}}) l_{\sigma(i)}(T_{\sigma(h)}).$$

Therefore,

$$|T_{\sigma(h)}| \geq l_{\sigma(0)}(T_{\sigma(h)}) \geq \prod_{i=1}^{N}(1 + \frac{S_{\sigma(i)} - S_{\sigma(i-1)}}{CS_{\sigma(i)}}) l_{\sigma(N)}(T_{\sigma(h)}) > 2C l_{\sigma(N)}(T_{\sigma(h)}).$$

And consequently,

$$l_{\sigma(N)}(T_{\sigma(h)}) \leq \frac{|T_{\sigma(h)}|}{2C} \leq \frac{1}{2}S_{\sigma(h)} \leq S_{\sigma(h)} - S_{\sigma(N)}.$$

This means that the firefighters available between turns $\sigma(N)$ and $\sigma(h)$ outnumber the unprotected vertices on level $\sigma(h)$. Hence, the strategy will win in at most $\sigma(h)$ turns. $\square$

Conversely, asymptotic behaviours cannot guarantee that an instance will be losing. Indeed, if $f_1 \geq |T_1|$, the instance is winning regardless of asymptotic behaviours. However, having selected asymptotic behaviours where the levels of the tree grow faster than the firefighter sequence, theorem 10 guarantees that some instances with those asymptotic behaviours will be losing.

Theorem 10 ([20])**.** *Let $(t_i) \in {\mathbb{N}^*}^{\mathbb{N}^*}$ and $(f_i) \in {\mathbb{R}^+}^{\mathbb{N}^*}$ be such that (t_i) is non-decreasing and tends towards $+\infty$. Then, $\sum \frac{f_i}{t_i}$ converges if and only if there exists a spherically symmetric tree T rooted in r such that:*

- $\exists N : \forall i \geq N, \frac{t_i}{2} \leq |T_i| \leq t_i$
- *the instance $(T, r, (f_i))$ is losing for* (Fractional) Firefighter*.*

Proof. 1) Suppose that $\sum \frac{f_i}{t_i}$ converges. Let M be such that $\sum_{i=M+1}^{+\infty} \frac{f_i}{t_i} < \frac{1}{4}$ and let $N > M$ be such that $t_N > 4S_M$. We choose $a_1 = t_N$, $a_i = 1$ for $2 \leq i \leq N$, and $a_i = \left\lfloor \frac{t_i}{\prod_{j=1}^{i-1} a_j} \right\rfloor$ for $i > N$. We will show that $T = T((a_i))$ is a solution.

Let us show by induction that $\forall i \geq N, \frac{t_i}{2} \leq |T_i| \leq t_i$. Note that

$$|T_N| = \prod_{j=1}^{N} a_j = t_N.$$

Assume that the result holds for $i-1$ where $i > N$:

$$\frac{t_{i-1}}{2} \leq \prod_{j=1}^{i-1} a_j \leq t_{i-1}.$$

Since $t_i \geq t_{i-1}$, we have $a_i \geq 1$. Hence,

$$a_i \leq \frac{t_i}{\prod_{j=1}^{i-1} a_j} \leq a_i + 1 \leq 2a_i.$$

So, $\frac{t_i}{2} \leq |T_i| \leq t_i$, and the result holds for all $i \geq N$.

Since T is spherically symmetric, the amount of fire that spreads to level n is $\max(0, F_n)$, where $F_n = |T_n|(1 - \sum_{i=1}^{n} \frac{f_i}{|T_i|})$. For $n > N$, we have:

$$\begin{aligned}
\sum_{i=1}^{n} \frac{f_i}{|T_i|} &= \frac{S_N}{t_N} + \sum_{i=N+1}^{n} \frac{f_i}{|T_i|} \\
&\leq \frac{S_M}{t_N} + \frac{1}{t_N}\sum_{i=M+1}^{N} f_i + 2\sum_{i=N+1}^{n} \frac{f_i}{t_i} \\
&< \frac{1}{4} + \frac{1}{4} + \frac{2}{4} = 1.
\end{aligned}$$

Hence, $F_n > 0$ for all n, and therefore the fire cannot be contained.

2) Conversely, if $\sum \frac{f_i}{t_i}$ diverges and $T = T((a_i))$ is such that $\exists N : \forall i \geq N$, $\frac{t_i}{2} \leq |T_i| \leq t_i$, then $\sum \frac{f_i}{|T_i|}$ also diverges. It follows that $F_n = |T_n|(1 - \sum_{i=1}^{n} \frac{f_i}{|T_i|})$ is negative above a certain rank. Hence, the fire is contained. $\square$

Corollary 10 ([20])**.** *Let $(t_i) \in \mathbb{N}^{*\mathbb{N}^*}$ and $(f_i) \in \mathbb{R}^{+\mathbb{N}^*}$ be such that (t_i) is non-decreasing and tends towards $+\infty$. Let $S_i = \sum_{1 \leq k \leq i} f_k$. If $S_i \to +\infty$ and $\frac{S_i}{t_i} \nrightarrow 0$, then $\sum \frac{f_i}{t_i}$ diverges.*

Proof. If $\sum \frac{f_i}{t_i}$ were convergent, it follows from theorem 10 that there would be a spherically symmetric tree T such that:

- $\exists N : \forall i \geq N, \frac{t_i}{2} \leq |T_i| \leq t_i$
- the instance $(T, r, (f_i))$ is losing for (Fractional) Firefighter.

It then follows from theorem 9 that $S_i \nrightarrow +\infty$ or $\frac{S_i}{|T_i|} \to 0$. Hence $S_i \nrightarrow +\infty$ or $\frac{S_i}{t_i} \to 0$. $\square$

It follows that for Fractional Firefighter, theorem 9 is weaker than proposition 24. Theorem 9 remains interesting for Firefighter and it gives an alternative winning method for Fractional Firefighter.

Remark 18. *Under the hypotheses of theorem 10, if $\sum \frac{f_i}{t_i}$ converges, we can create a losing instance $(T', r', (f_i))$ with $\forall i, |T_i| = t_i$ by adding $t_i - |T_i|$ leaves to level i for all i. We will have $|T_n| = t_n$ without adding leaves if and only if there exists a spherically symmetric tree with t_i vertices on level i for all i.*

Remark 19. *Remark 1.12 in [23] gives a sufficient condition for an instance to be losing for* Firefighter *in a general graph satisfying some growth condition, using a similar criteria to the convergence of $\sum \frac{f_i}{t_i}$. In general, both results cannot be compared. In our set-up however, their result can be seen as the particular case of* Firefighter *where $(t_i) = (\lambda^i)$ for some λ.*

3.5.2 Conjecture

This section describes my attempts at extending proposition 24 to include the case of Firefighter. After reducing the problem to the case of one firefighter per turn, two possible strategies are suggested, both coming close to solving the problem. It is my hope that they might eventually provide keys to a complete solution.

Conjecture 2. *Let $(T, r, (f_i))$ be an instance of* Firefighter *where T is a tree of infinite height. If $\sum_{i=1}^{+\infty} \frac{f_i}{|T_i|} > 1$, then the instance is winning.*

Let us start with a reduction:

Lemma 12 ([20])**.** *In order to show that instances of* Firefighter *where $\sum \frac{f_i}{|T_i|}$ diverges are winning, it is sufficient to prove it for instances with $f_i = 1$, $\forall i$.*

Proof. Given an instance $(T, r, (f_i))$ of FIREFIGHTER where $f_i = p \geq 2$, an equivalent instance is obtained by adding p_1 vertices on each edge between T_i and T_{i+1} and replacing f_i by $1, 1, \dots, 1$, f_i times, in the firefighter sequence. Repeating this process for each i gives an equivalent instance with one firefighter per turn, and the nature of $\sum \frac{f_i}{|T_i|}$ is unchanged. $\square$

Using this reduction, we may consider only instances with one firefighter per turn. A first candidate for a winning strategy is to select a level h, with h large enough, and protect at each turn i the vertex $v_i \in T_i$ with the highest number of descendants in T_h, as in the proof of theorem 9. In the hope of obtaining a contradiction, we may assume that this strategy is not winning in before turn h, which means, using the same notations as in the proof of theorem 9, that $\sum_{i=1}^{h} w_h(v_i) < |T_h|$. It follows from the pigeonhole principle that at each turn $i \leq h$, there is an unprotected vertex $v \in T_i$ such that $w_h(v) \geq \frac{l_{i-1}(T_h)}{l_{i-1}(T_i)}$. Hence, $w_h(v_i) \geq \frac{l_{i-1}(T_h)}{l_{i-1}(T_i)}$ and $1 > \frac{1}{|T_h|}\sum_{i=1}^{h} w_h(v_i) \geq \sum_{i=1}^{h} \frac{1}{|T_h|}\rho_{ih}$, where $\rho_{ih} = \frac{l_{i-1}(T_h)}{|T_h|}\frac{|T_i|}{l_{i-1}(T_i)}$. Since $\sum \frac{1}{|T_i|}$ diverges, we can choose h such that $\sum_{i=1}^{h} \frac{1}{|T_i|} > \frac{1}{\epsilon}$, for $\epsilon > 0$. It follows that $\frac{\sum_{i=1}^{h} \frac{1}{|T_i|}\rho_{ih}}{\sum_{i=1}^{h} \frac{1}{|T_i|}} < \epsilon$.

In other words, the barycentre of the ρ_{ih} with coefficients $\frac{1}{|T_i|}$ tends to 0. Since $\frac{|T_i|}{l_{i-1}(T_i)} \geq 1$, this implies that $\frac{l_{i-1}(T_h)}{|T_h|}$ tends to 0. Unfortunately, this does not guarantee that this strategy is winning. For this strategy to fail for a given h, we need it to protect vertices, the descendants of which form a greater proportion of T_h than of the T_j, for $i < j < h$. This can be done using chains which split into many vertices at level h, so that protecting a vertex of the chain at turn i protects a single vertex in each level T_j, but many vertices in T_h. The real problem is whether or not there exists a tree for which this strategy will fail for any h. I tried looking at trees where $|T_i| = \lfloor i\log(i)\rfloor$ and all vertices of T_i except one have exactly one child. This design creates long chains which split into many vertices at some point, but none of my attempts even came close to providing a working counter-example.

In the case of spherically symmetric trees, the coefficients ρ_{ih} are always equal to 1, which is why $\sum_{i=1}^{h} \frac{1}{|T_i|} > 1$ is sufficient in that case to guarantee a win. Indeed, in spherically symmetric trees, placing a firefighter on a vertex $v_i \in T_i$ protects the same proportion of each level T_j, $j > i$. The problem with the previous strategy is that we lose too much by selecting moves based only on their impact at level h with no regards for other levels. Thus, a refined idea is to select moves which protect a high proportion of not just one, but an infinite number of lower levels. At turn i, for each h, there is at least one vertex in T_i such that $w_h(v) \geq \frac{l_{i-1}(T_h)}{l_{i-1}(T_i)}$. Since T_i is finite, there is a vertex $v_i \in T_i$ which satisfies $w_h(v_i) \geq \frac{l_{i-1}(T_h)}{l_{i-1}(T_i)}$ for an infinite number of values of h. We can even choose v_i so that the set $H = \{h > i, w_h(v_i) \geq \frac{l_{i-1}(T_h)}{l_{i-1}(T_i)}\}$ satisfies $\sum_{h\in H} \frac{1}{|T_h|}$ is divergent. The following process is used to describe the resulting new strategy:

For any $A \subset \mathbb{N}^*$, we define the *measure* of A as $\mu(A) = \sum_{h\in A} \frac{1}{|T_h|}$. We now define the sequence (H_i) by induction with $H_0 = \mathbb{N}^*$, if $i \notin H_{i-1}$, then $H_i = H_{i-1}$ and if $i \in H_{i-1}$, then there is a vertex $v_i \in T_i$ such that $\mu(\mathcal{H}_{i,v_i}) = +\infty$, where $\mathcal{H}_{i,v} = \{h \in H_{i-1}, w_h(v) \geq \frac{l_{i-1}(T_h)}{l_{i-1}(T_i)}\}$, and we set $H_i = H_{i,v_i}$. The strategy is this: when $i \in H_{i-1}$, we protect v_i, so the labelling is updated via $\forall v \in T$, $l_i(v) = l_{i-1}(v) - \mathbb{1}_{v_i \trianglelefteq v}$; and when $i \notin H_{i-1}$, we skip the turn, so $l_i = l_{i-1}$. Note that $\forall i$, $\mu(H_i) = +\infty$, and for $i \in H_{i-1}$, as long as the fire is not contained, v_i is well defined, if not uniquely, since $H_{i-1} = \bigcup_{i\in T_i, l_{i-1}(v)=1} \mathcal{H}_{i,v}$. The induction process stops if and when the fire is contained.

Let $H = \{i \in H_{i-1}\}$. This strategy was designed to obtain the same advantage we had with spherically symmetric trees. It allows us to extract an infinite set of turns, $H \subset \mathbb{N}^*$, so that there is a sequence of moves $(v_i, i \in H)$ where $\forall i, j \in H, i < j, w_j(v_i) \geq \frac{l_{i-1}(T_j)}{l_{i-1}(T_i)}$. We call such a set H a *good set*. This is where disaster occurs: while H was built via an infinite sequence of extractions of subsets of infinite measure, H may not have infinite measure itself! If we could find a good set H with $\mu(H) > 1$, we would have a winning strategy. If the induction process above produces an H of measure less than 1, it basically means that we have skipped too many turns. The complement of H still has infinite measure, so we can try using the turns we skipped, reinitialising the process with $H_0 = \mathbb{N}^* \backslash H$. By repeating this, we can partition $\mathbb{N}^*$ into good sets. Alas, there is still no guarantee that one of them will have a measure greater than 1.

I have not yet found an example that resists either of the two strategies described in this section. A counter-example to conjecture 2 would need to resist both of them.

3.5.3 Online firefighting on trees with linear level growth

In the previous section, proposition 24 gives a winning strategy for online Fractional Firefighter in cases where $\sum \frac{f_i}{|T_i|} > 1$. However, theorem 9 is limited to the offline case, as the winning strategy requires the player to be able to compute $\sigma(h)$ from the start. In this section, we give a result which works for online Firefighter in the case of rooted trees (T, r) where the number of vertices per level increases linearly, i.e. $|T_i| = \mathcal{O}(i)$. We say that such a tree has *linear level growth*.

Remark 20. *The linear level growth property of T remains if we choose a different root r'. Indeed, if d is the distance between r and r', the set of vertices at distance i from r' is included in $\bigcup_{j=i-d}^{i+d} T_j$, the cardinal of which is $\mathcal{O}(i)$.*

Theorem 11 ([20])**.** *There is an online algorithm for instances $(T, r, (f_i))$ of* Firefighter *where T has linear level growth, such that if some non-zero periodic sequence is weaker than (f_i), the fire will be contained.*

The proof of theorem 11 will use the following lemma:

Lemma 13 ([20])**.** *For any real number $a > 1$, $\lim_{n \to +\infty} \prod_{j=1}^{n} \frac{ja-1}{ja} = 0$.*

Proof. We have $\ln \prod_{j=1}^{n} \frac{ja-1}{ja} = \sum_{j=1}^{n} \ln(1 - \frac{1}{ja})$ and since $\sum_{j=1}^{n} \frac{1}{ja} \to +\infty$, we have $\sum_{j=1}^{n} \ln(1 - \frac{1}{ja}) \to -\infty$. Hence, $\prod_{j=1}^{n} \frac{ja-1}{ja} \to 0$. □

We can now prove theorem 11:

Proof. Since T has linear level growth, let C be such that $\forall i, |T_i| \leq Ci$. Without loss of generality, we assume $C > 1$. That a non-zero periodic sequence is weaker than (f_i) means that $(\mathbb{1}_{n|i}) \preceq (f_i)$ for all n greater than some m. First, we will give an offline strategy to contain the fire with one firefighter every n turns. Then, we will show that online instances with $(\mathbb{1}_{n|i}) \preceq (f_i)$ for an n known to the player are winning. Finally, we will describe an online winning strategy when such a $(\mathbb{1}_{n|i})$ is unknown.

Given an integer n, let us first consider the instance $(T, r, (\mathbb{1}_{n|i}))$. It follows from lemma 13 that there exists an integer N such that $\prod_{j=1}^{N} \frac{Cnj-1}{Cnj} < \frac{1}{2Cn}$. Let $h(n) = 2nN$. A winning strategy for this offline instance is obtained by protecting at turn nj the unprotected vertex of T_{nj} with the highest number of descendants in level $h(n)$. Since $|T_{nj}| \leq Cnj$, the remaining number of unprotected vertices in $T_{h(n)}$ is reduced by at least $\frac{1}{Cnj}$ of its previous value. So the number of unprotected vertices of $T_{h(n)}$ remaining after nN turns is less than $|T_{h(n)}| \prod_{i=1}^{N} \frac{Cnj-1}{Cnj} \leq \frac{|T_{h(n)}|}{2Cn} \leq N$. Since N firefighters remain to be placed between turns N and $h(n)$, the strategy is winning in at most $h(n)$ turns.

If the player knows in advance that $(\mathbb{1}_{n|i}) \preceq (f_i)$ for a given n, the above strategy can be adapted using lemma 9.

In the general case, assume that $(\mathbb{1}_{n|i}) \preceq (f_i)$ for some n, but the player does not know which n. The online strategy proceeds as follows: we initially play as though under the assumption that $(\mathbb{1}_{n_0|i}) \preceq (f_i)$ with $n_0 = 100$. If the fire is not contained by turn $h(n_0)$, or later on by turn $h(n_k)$, we choose $n_{k+1} = h(n_k)\left(\left\lceil S_{h(n_k)} \right\rceil + 1\right)$. We now assume that $(\mathbb{1}_{h(n_k)|i}) \preceq (f_i)$. It follows that after cancelling the first $h(n_k)$ terms of (f_i), i.e., replacing f_ℓ with 0 for $\ell \leq h(n_k)$, the resulting sequence is stronger than $(\mathbb{1}_{n_{k+1}|i})$. So we can consider that the first $h(n_k)$ turns were wasted and follow the strategy for n_{k+1} until turn $h(n_{k+1})$. Eventually, this strategy will win when n_k is large enough. $\square$

3.6 Future research

By considering a general sequence of number of firefighters available at each turn in (Fractional) Firefighter, we obtained three independent research questions: the online version, the separating problem, and determining which infinite instances are winning based on asymptotic behaviours.

After introducing the online version of (Fractional) Firefighter on trees, we provided initial results for the finite case. So far, our results outline the potential of this approach and suggest many open questions. To our knowledge, theorem 5 is the first non-trivial competitive (and also approximation) analysis for Fractional Firefighter and a first question would be to investigate whether a better competitive ratio can be obtained for Fractional Firefighter in finite trees. Although the case of trees is already challenging, the main open question will be to study online (Fractional) Firefighter problem in other classes of finite graphs.

As far as we know, the second question has never before been considered. The existence of a separating tree for any two given firefighter sequences seems very hard in general. Spherically symmetric trees provide convenient examples of separating trees since they allow us to ignore the playing strategy. This allowed us to express the problem in terms of the targeting game, which completely hides the structure of the tree. An interesting question will be to investigate whether the existence of a separating tree implies that of a spherically symmetric separating tree. So far, we only considered the case where one of the sequences is weaker than the other. The general case remains fully open.

We have shown that some conditions on the asymptotic behaviours of the firefighter sequence vs. the tree growth guarantee that the instance is winning. Yet, other conditions guarantee the existence of losing instances. We conjectured that all Firefighter instances where $\sum \frac{f_i}{|T_i|}$ diverges are winning.

Finally, note that the question of approximating (Fractional) Firefighter in finite trees for a general firefighter sequence is also an important research direction that, to our knowledge, remains uninvestigated.

Chapter 4

Conclusion

4.1 Ongoing and future works

While specific research questions have already been mentioned at the end of each chapter, the following problems combine results and ideas from different chapters.

A long term objective common to the first two chapters of this thesis is the characterisation of partially looped or non-looped copwin graphs. While chapter 2 only considers totally looped graphs, an important research direction is to extend the results to the partially looped case. Any partially looped copwin graph must have a vertex u dominated in the sense that there is a vertex v such that $N(u) \subset N(v)$, otherwise the robber cannot be trapped anywhere. However, removing u does not necessarily produce a copwin graph. Yet, if we remove u and add a loop on v that can only be used by the cop, then the resulting graph is copwin. Removing u may be viewed as applying a one-point retraction where the displaced point and its image need not be adjacent. It follows that dismantlability via those one-point retractions is a necessary condition for partially looped graphs to be copwin. Following the same reasoning as in chapter 2, we can deduce that the group of circulations for partially looped copwin graphs is generated by cyclic flows based on cycles of lengths 1 to 4. Candidates satisfying this condition include grids or partial grids with loops. Unfortunately, the case of grids is already extremely complex, as illustrated by the partially looped $2 \times n$ grids considered in chapter 1. Reaching a characterisation probably requires a more complete understanding of the strange behaviours which occur on those grids. The step above grids would be tilings with both triangular and quadrilateral cells.

Since the online versions of the firefighter problem seem very promising, a similar idea would be to introduce an online version of the game of cops and robbers. For instance, we could play the game with n cops while the adversary reveals at each turn how many cops can make a move. We could also let the adversary decide specifically which cops are allowed to move. Both of these online variants could be enhanced by allowing cops to make several moves in one turn. To my knowledge, no one has studied these variants yet.

During Bertrand Jouve's last visit to Melbourne in March 2018, we started examining a new type of firefighting game. Bertrand's stay coincided with the visit of a Corsican firefighter who also worked for Geosafe. He explained to us the differences in firefighting management in Australia and in Corsica. When a fire starts in the Australian bush, a major issue is to detect it and get there in time before the fire burns so hot that nothing can be done to control it. However, when a fire starts in Corsica, thanks to having firefighting resources spread out all over the island, they can always reach it in time. Unfortunately, those resources are limited, and if there are too many fires, they need to prioritise the most dangerous ones. In fact, it is important for them do define a strict policy for making such decisions, especially when human lives may depend on it. Such a policy can be described as a classic online travelling salesman problem with time windows, generalising a problem studied in [5]: a time window opens whenever a fire starts and closes when it burns too hot to be contained. The extra condition is that all the time windows have a width greater than the time it takes to travel from any point in the space considered. The time windows also have a weight describing the importance of stopping that fire. The initial results that we have obtained so far seem promising and should lead to a publication in the foreseeable future.

4.2 Acknowledgements

I would like to thank my supervisors Marc Demange and Graham Clarke for guiding me throughout my PhD, giving kind support while keeping me on track and proofreading my writings. All my work on the firefighter problem was done in collaboration with Pierre Coupechoux, Marc Demange and Bertrand Jouve. Pierre and Bertrand travelled to Melbourne with funding from Geosafe Project H2020 and from the French Embassy. I would also like to thank Professor Dr. Dieter Rautenbach and Professor Dr. Nancy Clarke, who examined this thesis and gave excellent comments and suggestions.

Bibliography

[1] D. Adjiashvili, A. Baggio, and R. Zenklusen. Firefighting on trees beyond integrality gaps. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 2364–2383, 2017.

[2] M. Aigner and M. Fromme. A game of cops and robbers. *Discrete Applied Mathematics*, 8(1):1–12, 1984.

[3] E. Anshelevich, D. Chakrabarty, A. Hate, and C. Swamy. Approximability of the firefighter problem - computing cuts over time. *Algorithmica*, 62(1-2):520–536, 2012.

[4] G. Ausiello and L. Becchetti. On-line algorithms. In V. Th. Paschos, editor, *Paradigms of Combinatorial Optimization: Problems and New Approaches, Vol. 2*, chapter 15, pages 473–509. ISTE - WILEY, London - Hoboken, 2010.

[5] G. Ausiello, M. Demange, L. Laura, and V. Paschos. Algorithms for the on-line quota traveling salesman problem. *Information Processing Letters*, 92(2):89–94, 2004.

[6] C. Bazgan, M. Chopin, M. Cygan, M.R. Fellows, F.V. Fomin, and E.J. van Leeuwen. Parameterized complexity of firefighting. *Journal of Computer and System Sciences*, 80(7):1285–1297, 2014.

[7] C. Bazgan, M. Chopin, and B. Ries. The firefighter problem with more than one firefighter on trees. *Discrete Applied Mathematics*, 161(7-8):899–908, 2013.

[8] A. Bonato, P. Golovach, G. Hahn, and J. Kratochvíl. The capture time of a graph. *Discrete Mathematics*, 309(18):5588–5595, 2009.

[9] A. Bonato, M.E. Messinger, and P. Pralat. Fighting constrained fires in graphs. *Theoretical Compututer Science*, 434:11–22, 2012.

[10] A. Bonato and R. Nowakowski. *The game of cops and robbers on graphs*, volume 61. American Mathematical Society, Rhode Island, 2011.

[11] R. Boulet, E. Fieux, and B. Jouve. Simplicial simple-homotopy of flag complexes in terms of graphs. *European Journal of Combinatorics*, 31(1):161–176, 2010.

[12] M. Boyer, S. El Harti, A. El Ouarari, R. Ganian, G. Hahn, C. Moldenauer, I. Rutter, B. Thériault, and M. Vatshelle. Cops-and-robbers: remarks and problems. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 85:107–128, 2013.

[13] M.H. Breitner. The genesis of differential games in light of isaacs contributions. *Journal of Optimization Theory and Applications*, 124(3):523–559, 2005.

[14] G. Brightwell and P. Winkler. Gibbs measures and dismantlable graphs. *Journal of combinatorial theory, series B*, 78(1):141–166, 2000.

[15] L. Cai, E. Verbin, and L. Yang. Firefighting on trees: (1-1/e)-approximation, fixed parameter tractability and a subexponential algorithm. In *Algorithms and Computation, 19th International Symposium, ISAAC 2008, Gold Coast, Australia, December 15-17, 2008. Proceedings*, pages 258–269, 2008.

[16] J. Chlebíková and M. Chopin. The firefighter problem: Further steps in understanding its complexity. *Theoretical Compututer Science*, 676:42–51, 2017.

[17] N. Clarke, S. Finbow, and G. MacGillivray. A simple method of computing the catch time. *Ars Mathematica Contemporanea*, 7(2), 2013.

[18] N.E. Clarke, S. Finbow, S.L. Fitzpatrick, M.-E. Messinger, and R.J. Nowakowski. Seepage in directed acyclic graphs. *Australasian J. Combinatorics*, 43:91–102, 2009.

[19] P. Coupechoux, M. Demange, D. Ellison, and B. Jouve. Online firefighting on trees. In *ISCO 2018*, 2018.

[20] P. Coupechoux, M. Demange, D. Ellison, and B. Jouve. Firefighting on trees. *Theoretical Computer Science*, 2018, In press.

[21] R. Diestel. *Graph theory {graduate texts in mathematics; 173}*. Springer-Verlag Berlin and Heidelberg GmbH & amp, 2000.

[22] A. Dochtermann. Hom complexes and homotopy theory in the category of graphs. *European Journal of Combinatorics*, 30(2):490–509, 2009.

[23] D. Dyer, E. Martínez-Pedroza, and B. Thorne. The coarse geometry of Hartnell's firefighter problem on infinite graphs. *Discrete Mathematics*, 340(5):935–950, 2017.

[24] O.N. Feldheim and R. Hod. 3/2 firefighters are not enough. *Discrete Applied Mathematics*, 161(1-2):301–306, 2013.

[25] E. Fieux and J. Lacaze. Foldings in graphs and relations with simplicial complexes and posets. *Discrete Mathematics*, 312(17):2639–2651, 2012.

[26] S. Finbow, A. King, G. Macgillivray, and R. Rizzi. The firefighter problem for graphs of maximum degree three. *Discrete Mathematics*, 307(16):2094–2105, 2007.

[27] S. Finbow and G. MacGillivray. The firefighter problem: a survey of results, directions and questions. *Australasian Journal of Combinatorics*, 43(6):57–77, 2009.

[28] P. Fogarty. *Catching the Fire on Grids, PhD thesis*. University of Vermont, 2003.

[29] F.V. Fomin, P.A. Golovach, and D. Lokshtanov. Guard games on graphs: keep the intruder out! In *International Workshop on Approximation and Online Algorithms*, pages 147–158. Springer, 2009.

[30] F.V. Fomin, P. Heggernes, and E.J. van Leeuwen. The firefighter problem on graph classes. *Theoretical Compututer Science*, 613(C):38–50, February 2016.

[31] M.R. Garey and D.S. Johnson. *Computers and intractability*, volume 29. wh freeman New York, 2002.

[32] T. Gavenčiak. Cop-win graphs with maximum capture-time. *Discrete Mathematics*, 310(10):1557–1563, 2010.

[33] T. Gavenčiak. Cop-win graphs with maximum capture-time. *Discrete Mathematics*, 310(10–11):1557 – 1563, 2010.

[34] A. Grigor'yan, Y. Lin, Y. Muranov, and S. Yau. Homotopy theory for digraphs. *arXiv preprint arXiv:1407.0234*, 2014.

[35] Stephen G. Hartke. Attempting to narrow the integrality gap for the firefighter problem on trees. In *Discrete Methods in Epidemiology*, pages 225–232, 2004.

[36] B. Hartnell. Firefighter! An application of domination. 1995. presented at the 10th Conference on Numerical Mathematics and Computing, University of Manitoba in Winnipeg, Canada.

[37] B. Hartnell and Q. Li. Firefighting on trees: How bad is the greedy algorithm? *Congressus Numerantium*, pages 187–192, 2000.

[38] A. Hatcher. *Algebraic topology*. Cambridge University Press, 2002.

[39] Y. Iwaikawa, N. Kamiyama, and T. Matsui. Improved approximation algorithms for firefighter problem on trees. *IEICE Transactions on Information and Systems*, E94.D(2):196–199, 2011.

[40] J. Kling and B. Horwitz. *Chess Studies: Or, Endings of Games*. CJ Skeet, 1851.

[41] F. Lehner. Firefighting on trees and Cayley graphs. *ArXiv e-prints, (arXiv:1707.01224v1 [math.CO])*, July 2017.

[42] L. Lu and X. Peng. On meyniel's conjecture of the cop number. *Journal of Graph Theory*, 71(2):192–205, 2012.

[43] R. Lyons and Y. Peres. *Probability on Trees and Networks*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2017.

[44] G. MacGillivray and P. Wang. On the firefighter problem. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 47:83–96, 2003.

[45] M.E. Messinger. Average firefighting on infinite grids. *The Australasian Journal of Combinatorics*, 41:15–28, 2008.

[46] S. Neufeld and R. Nowakowski. A game of cops and robbers played on products of graphs. *Discrete Mathematics*, 186(1):253–268, 1998.

[47] R. Nowakowski and P. Winkler. Vertex-to-vertex pursuit in a graph. *Discrete Mathematics*, 43(2):235–239, 1983.

[48] T. Parsons. Pursuit-evasion in a graph. *Theory and Applications of Graphs*, pages 426–441, 1976.

[49] A. Quilliot. Problèmes de jeux, de point fixe, de connectivité et de représentation sur des graphes, des ensembles ordonnés et des hypergraphes. *Thèse de doctorat détat, Université de Paris VI, France*, 1978.

[50] A. Scott and B. Sudakov. A bound for the cops and robbers problem. *SIAM Journal on Discrete Mathematics*, 25(3):1438–1442, 2011.

[51] D. Stanley and B. Yang. Fast searching games on graphs. *Journal of combinatorial optimization*, 22(4):763–777, 2011.

[52] M. E. Talbi and D. Benayat. Homology theory of graphs. *Mediterranean Journal of Mathematics*, 11(2):813–828, 2014.

[53] P. Wang and S.A. Moeller. Fire control on graphs. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 41:19–34, 2002.

[54] B. Yang, D. Dyer, and B. Alspach. Sweeping graphs with large clique number. *Discrete Mathematics*, 309(18):5770–5780, 2009.

Publications

Here is my list of publications. The first three were produced during my undergraduate studies, as I studied game theory with professor Michel Rudnianski. The fourth was produced just before the start of my thesis, as I studied algebraic graph theory with Marc Demange and Cerasela Tanasescu. While the fifth was produced during my PhD, its topic is also algebraic graphs, so the contents were not included in my thesis. The final three are the meat and bones of this thesis.

- Game Theory:

  D. Ellison, M. Rudnianski. Is Deterrence Evolutionarily Stable?. *Advances in Dynamic Games and Their Applications*, Birkhauser, Boston 2009

  D. Ellison, M. Rudnianski. Compliance Pervasion and the Evolution of Norms: the Game of Deterrence Approach. *Contributions to Game Theory and Management*, 2 (2009), 383-414

  D. Ellison, M. Rudnianski. Playability Properties in Games of Deterrence and Evolution in the Replicator Dynamics. *Contributions to Game Theory and Management*, 6 (2013), 115-133

- Algebraic Graphs:

  D. Ellison, R. Marinescu-Ghemeci, C. Tanasescu. G-graphs Characterisation and Incidence Graphs. ArXiv preprint arXiv:1405.3102. 2014 May 13, to be submitted
  B. Mourad, M. Badaoui, A. Bretto, D. Ellison. On Constructing Expander Families of G-graphs. Recently accepted, *Ars Mathematica Contemporanea*

- Games on Graphs:

  D. Ellison. The Impact of Loops on the Game of Cops and Robbers on Graphs. Under review, final round, *Journal of Discrete Mathematics*

  P. Coupechoux, M. Demange, D. Ellison, B. Jouve. Online Firefighting on Trees. Recently accepted, *Proceedings of International Symposium on Combinatorial Optimization* 2018

P. Coupechoux, M. Demange, D. Ellison, B. Jouve. The Firefighter Problem on Trees. Under review, final round, *Theoretical Computer Science*