On Discrete and Geometric Firefighting

Barbara Anna Schwarzwald
geboren in Bonn

Dissertation
zur Erlangung des Doktorgrades (Dr. rer. nat)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

Mai 2020

1. Gutachter: Prof. Dr. Rolf Klein
2. Gutachterin: Prof. Dr. Anne Driemel

Tag der mündlichen Prüfung: 10.09.2020
Erscheinungsjahr: 2021

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

“The most important step a man can take is not the first one. It’s the next one. Always the next step.”
Brandon Sanderson, Oathbringer

I would like to dedicate this thesis to Simon, who always helped me take the next step.

Abstract

Wildfires ravaging forests around the globe cost lives, homes and billions in damages every year, which motivates the study of effective firefighting. In the area of theoretical computer science, several different models inspired by firefighting have been established and studied to find efficient firefighting strategies.

In Hartnell's fire fighter problem [44], a fire burns through the vertices in a graph in rounds: in each round, the fire spreads from each burning vertex to all adjacent vertices. A firefighter is tasked with protecting as many vertices of the graph as possible by blocking a single vertex each round. While a plethora of results have been obtained with respect to specific graph classes or variations of the firefighter's power, the modus operandi of the fire rarely changes.

We study a new model generalizing the one used by Hartnell to better simulate the varying speeds, at which a fire spreads through different terrain types, by incorporating a fire resistance and energy for each vertex. We present an efficient algorithm to track the fire propagation in a given graph $G = (V,E)$ in time $O(|E| \cdot |V|)$, that is particularly efficient in graphs with bounded vertex degree, where its runtime lies in $O(|V| \log |V|)$. We also obtain polynomial-time algorithms for two problems regarding protecting a set of vertices along the boundary of a hexagonal cell graph. We show *NP*-completeness for an inverted third problem, where the goal is instead to ignite a set of target cells given a set of starter cells. We also examine the unique features of the model and propose a number of new questions utilizing them.

In the second part of this thesis we focus on geometric firefighting as introduced by Bressan [14]. In that model, the fire burns a region of the Euclidean plane that grows over time with unit speed, and has to be contained by building barrier curves with some building speed v. In the original problem, the exact necessary building speed to contain the fire is not known, as a gap remains between the best known strategy for a speed of $v > 2$ and a lower bound of $v > 1$. The difficulty in closing this gap seems to lie with the following question: *To contain a fire, should one build an enclosing barrier at maximum speed, or is it better to invest some time in building extra delaying barriers that will not be part of the final enclosure but can slow the fire down during construction?*

To get a step closer to the answer to this question, we mainly study a variant of the original problem, where a fire spreading at unit speed according to the L_1-metric is to be contained in an open half-plane. Towards that goal, the firefighter is allowed to build one infinite enclosing barrier along the x-axis, and vertical delaying barriers attached to it. We prove that at least a building speed of

$v > 1.\overline{6}$ is necessary to contain the fire in this variant, while providing a strategy that suceeds for a speed $v > 1.8772$. We also study some smaller variants of both this and the original problem.

In the final part of this thesis, we study the Minimum Enclosing Ball problem in high dimensions: given a set of n points in the d-dimensional Euclidean space, find the ball of minimum radius containing all points. This is a classic clustering problems and has been studied extensively in the past, often together with its generalization, the Euclidean k-center problem. Among the known results are polynomial-time algorithms to obtain optimal solutions for fixed k and d [64, 82], polynomial-time $(1+\varepsilon)$-approximation algorithms for $k = 1$, but arbitrary d [7, 61] as well as – most recently – a polynomial-time algorithm for instance with rational coefficients [74]. However, it can not be solved in polynomial time for general d. We provide a simple gradient-descent based $(1+\varepsilon)$-approximation algorithm, that runs in time $O\left(nd\frac{1}{\varepsilon}\right)$ and improves on the similar core-set based approximation algorithms.

Acknowledgements

I would like to thank several people, who made this thesis possible by supporting me in one way or the other during these nearly five years.

First and foremost, I wish to thank my supervisor Rolf Klein for his invaluable insight and guidance. The enthusiasm he displayed in teaching inspired me to start this project, and was equally contagious during research. The results in this thesis all arose from our fruitful discussions, which I never left without a new approach how to tackle the current hurdle. Besides his immensely helpful scientific expertise, I also got to experience him as simply one of the kindest persons I have ever met.

I also want to express my thanks to my other co-authors, namely Herman Haverkort, Sang-Sub Kim, David Kübel, Elmar Langetepe, Christos Levcopoulos, Andrzej Lingas and Jörg-Rüdiger Sack. In particular, I want to thank David Kübel and Elmar Langetepe, who joined Rolf Klein and me in countless “Research Jam Sessions”.

I want to give special thanks to Heiko Röglin for seamlessly welcoming me into his department after Rolf Klein’s retirement. I also greatly appreciate the many nice colleagues I got to know in both department I and V, with whom I shared many interesting conversations over the years. They made me enjoy every day in the office with regular cakes (with or without occasion), occasional board game afternoons and the simple, but most important, every day good mood. Most notably, I want to thank my officemates David and Carsten, who probably kept my plants from dying more than once, and always found the time to discuss a short question on the whiteboard.

Last but not least, I want to thank my family, my friends and, in particular, my husband Simon for their love and encouragement.

Table of contents

List of figures

Chapter 1

Introduction

When hearing the word *algorithm*, a lot of people think of Google showing them ads based on their last searches, analytic companies influencing elections using facebook groups, or self-driving cars deciding to brake. However, the fundamental idea is just to train a computer to help us solve complex problems efficiently. Big Data Analysis and Artificial Intelligence, as often in the news today, are just a small portion of those problems solved in computer science.

While inspired by problems and questions in the real world, theoretical computer scientists study somewhat simplified abstractions of these problems. Finding good abstractions helps us to understand what is at the core of the questions we ask about the world, and sometimes leads us to discover that two problems, that seem very different, are fundamentally the same. They also allow us to mathematically prove the correctness or efficiency of our solutions, an endeavour that is likely to fail when trying to consider each and every parameter of our infinitely complex reality, that could affect any given problem.

In the field of computational geometry, the problems studied are often inspired by movement in the real world. It can be searching a way out of any maze or finding out that it does not have an exit [1], finding the fastest route by car from home to work [28], or finding the closest supermarket or post office on a city map and where to put another one to attract the most customers [5]. While describing these problems is often intuitive, solving them can be surprisingly complex.

The problems examined in this thesis are inspired by firefighting. Forest wildfires that ravage especially Australia and North America on a regular basis cause billions of dollars of damages every year and burn hundreds of homes [32], sometimes even threatening larger settlements and urban areas. As fighting these fires is costly, dangerous and exhausting for the firefighters involved, one naturally wants to do it in the most efficient way possible. But before ever designing an algorithm that might help firefighters coordinate their efforts, one must first model the problem and analyse possible solutions.

When finding abstract models for any problem, but especially for those inspired by nature, the balance between simplicity and power of expression is of utmost importance. On the one hand, the model must be sufficiently detailed to represent the problem adequately. Otherwise, results obtained

for the abstract model have no bearing in reality. On the other hand, it should be kept as simple as possible to still be able to obtain reasonable results and avoid both unnecessary complexity and – in the worst case - undecidable problems. Hence, it does not surprise that different firefighting models have been discussed in the scientific community. They can mostly be split into two distinct groups: *discrete* firefighting in graphs and *continuous* firefighting in the Euclidean plane, also called *geometric* firefighting.

1.1 Methods and Prerequisites

From here on out, this thesis assumes general familiarity with basic concepts of theoretical computer science. In the following, we will give a short overview of the prerequisites utilized and where to best get an introduction to them, if any are unfamiliar.

This includes basic complexity theory to compare the efficiency of algorithms presented within this thesis, more specifically O-, Ω and Θ-Notation as well as the complexity classes P and NP. A good introduction into these topics can be found in *Computational Complexity - A Modern Approach* by Arora and Barak [3].

Especially in Chapter 2, we will rely on knowledge of basic graph theory and algorithms. This includes both the definitions and basic properties of graphs and common graph classes, as well as basic search algorithms like Breadth- and Depth-First-Search or Dijkstra's shortest path algorithm. These topics are well covered by a number of books, not least in *Introduction to Algorithms* by Cormen et al. [24] and *Graph theory with applications* by Bondy et al. [13].

For a comprehensive introduction to algorithms from the area of computational geometry specifically, one can also study *Computational Geometry* by de Berg et al. [25] or *Algorithmische Geometrie* by Klein [52].

The analyses provided in this thesis additionally utilize fundamental mathematical techniques, for example linear algebra to solve linear recurrences. For a good foundation of mathematical background, refer to *Concrete Mathematics* by Graham et al. [41].

1.2 Discrete Firefighting

The classic firefighter problem was first introduced by Hartnell at a Conference in 1995 [44] and is defined on a graph.

Hartnell's Firefighter
Input: *A graph $G = (V,E)$ and a* burning *starting vertex $s \in V$.*
Consider the following game played in rounds: In each round, the firefighter is allowed to mark a vertex $v \in V$ that is not burning as 'protected'. *Then, the fire spreads to all non-protected vertices adjacent to at least one burning vertex. All burning or protected vertices stay so until the end of the game. The game is played until there are no more non-protected vertices adjacent to a burning vertex.*

Goal: *Find a sequence of vertices to protect that minimizes the number of burned vertices when the game ends.*

Finbow and MacGillivray provide an excellent comprehensive survey of results and related research up to 2009 [33]. Nevertheless, we will give an overview here, also including some more recent developments.

The firefighter problem is trivially solvable for paths and cycles and, by a natural extension of that, in graphs of maximum degree three, as long as the starting vertex has degree at most two [34]. In contrast, the firefighter problem was proven to be *NP*-complete even in fairly simple graph classes like bipartite graphs [63], trees of maximum degree three [34], and cubic graphs [51].

It can also not be approximated to a factor of $n^{(1-\varepsilon)}$ in general graphs [2] and only trees are known to allow a constant-factor $(e/(e-1))$ approximation algorithm [23]. An interesting variation occurs when considering a variant with a budget b, where the firefighter is allowed to protect not one, but b many vertices in each round. In that case, maximizing the number of saved vertices is still *NP*-complete. Minimizing the budget necessary to save a specific subset of vertices however can be done within an approximation factor of $O(\sqrt{n})$ [2].

While studying general graphs, bipartite graphs or trees are natural points of view for graph theorists, they do not actually represent scenarios of a spreading fire very well. Of course, the same model also applies to other things that can spread - like a disease, a computer virus or a viral fake news story. Where the vertices represent people or computers, the edges represent physical or virtual contact. The protected vertices are then vaccinated people, computers with an updated antivirus program or people reached by a fact-checked counter argument. Especially with the last case as motivation, it is also valuable to consider variants where the protection is spreading in the same manner as the "fire" (in that case the fake news story) itself, as done by Anshelevich et al. [2]. In the context of spreading diseases, it is reasonable to consider nondeterministic variants, in which the "fire"(in this case the disease) spreads along each edge with a certain probability each round [78]. However, there already has been a lot of research on the propagation of diseases or computer viruses in the network and medical community. Motivated by real world properties of viruses and the typical connectivity of populations, they focus mostly on scale-free networks or small-world networks and use varied probabilistic models (for examples, see [27, 72, 84]).

For an actual fire however, it is more natural to consider graph classes, that can represent an area through which the fire spreads, like grid graphs. These are infinite planar graphs whose faces form a regular tiling in some Euclidean space. In infinite graphs, the game only ends if the fire is contained, i. e., completely surrounded by protected vertices. So before optimizing solutions, the first question to answer is whether the fire can be contained at all, and what budget b per round is necessary to do so.

To contain a fire in a d-dimensional square grid graph, a budget of $2d-1$ is sufficient and can be used to contain the fire in only two rounds with two vertices burned. In fact, for every dimension but 2, this budget is optimal. For the special case of 2, the fire can also be contained with a minimum budget of 2. In that case, however, an optimal strategy takes eight rounds and leaves 18 vertices

burned [26, 69]. This special behaviour of the low dimension is repeated when considering multiple fires breaking out at once. For $d = 2$, these can still be contained with a budget of 2 [36]. For $d \geq 3$, however, for any budget b there is a finite number of fires that can not be contained at all [26].

Besides the square grid, other natural grid types of varying regular vertex degree k have of course also been studied [36, 68] requiring a similar small constant optimal budget of $\lceil \frac{k}{2} \rceil$ in dimension 2.

1.2.1 Related Problems

Aside from questions focused around fighting the fire, one can also pose questions in the other direction: If we are allowed to set one vertex on fire each round, how many rounds r does it take at minimum until all vertices are burned by our actions and the subsequent spread. This value r is known as the burning number as introduced by Bonato et al. [12] and reflects the vulnerability of a graph to spreading fire.

Two closely related problems that have a certain similarity to firefighting are cop-and-robber games as well as the lion-and-man problem [11]. Belonging to a bigger family of pursuit-evasion games, they usually feature one or more pursuers (cops, lions), who are trying to catch a target (robber, man) moving through a graph. Depending on which party knows the others strategy and the number of pursuers, winning strategies for either party can be found on some graph classes [70].

Berger et al. [10] discuss a variant of the lion-and-men problem on grid graphs with a particularly close relation to the firefighting problem. In that variant, the position of the man is assumed to be unknown to the lions. Thus, they want to systematically prowl through the grid in a way that they catch him no matter where and how he moves. If the man could be at a vertex v at time t, at the next time step he could have moved to any adjacent vertex as long as there was no lion waiting for him. So the possible position of the man spreads similar to the fire. However, in contrast to the firefighter problem, as the lions move around, protected vertices become unprotected again and previously "burning" vertices can become protected.

1.2.2 Our Contribution

While the current research of firefighting in graphs varies with respect to the graph classes or power of the firefighters studied, the modus operandi of the fire stays nearly identical: it always spreads uniformly in all directions. In reality however, a fire spreads at varying speeds due to differences in wind, or vegetation and other terrain features.

In other communities, models have therefore been developed to predict a fire's propagation rate in a given terrain. To make the forecast as realistic as possible, some models incorporate thermodynamic or chemical parameters as well as weather conditions including wind speed and direction. Some of the models are capable to distinguish between fires at different heights such as ground fires and crown fires. Pastor et al. [71] survey a number of theoretical and (semi-)empirical models. One drawback they mention is that most models reduce the fire front to a single line, whereas in practise a whole

combustion zone should be taken into account. A drawback which Knight and Coleman [59] bring forward, is that most of these models assume that each point of fire propagates independently of its neighbours.

We introduce a new model, that generalizes the classical firefighting model in graphs with a focus on a more varied spread of the fire by giving energy and resistance values to vertices of the graph. The aim is to provide a still relatively simple theoretical framework, that is capable of simulating varied terrains and other parameters. This allows the fire to spread at different rates depending on the terrain, and can have multiple neighbours of a vertex affect its ignition, addressing the concern of Knight and Coleman mentioned above.

As this is a new model, we subsequently discuss its general properties and how it relates to the classic firefighting model and other related models. We also investigate the potential power lying in the adaptability of the generalized model. On the fundamental algorithmic side, we give an efficient algorithm to calculate the fire propagation in a graph with given energy and resistance values.

We also consider a series of problems related in a graph based on hexagonal cells, in which the goal is to protect a specific vertex subset representing a village from an approaching fire. Similar to questions like the burn number of graphs, we also study a problem where the goal is inverted and a set of target cells must be burned by igniting cells from a set of source cells.

1.3 Continuous Firefighting

A continuous version of the firefighter problem was introduced by Bressan in 2007 [14], where the fire is modelled as a subset of the Euclidean plane, i. e. a region of the plane that is on fire. Starting with a set R_0 at time $t = 0$, the set grows continuously over time according to some spreading function as the fire spreads through the plane. The goal is then to contain the fire within barrier curves serving a similar function as blocked vertices in the discrete problem. They are impassable to the fire, can not be moved or destroyed after building and must be built dynamically while the fire is spreading. The length of the barriers built is limited by a building speed v, i. e., the length of barriers built may never exceed $v \cdot t$ at any time t. These barriers represent forest parts preventively soaked in water to protect them from igniting or firebreaks created by deliberately cutting away strips of forest. The main task is to find strategies that minimize the building speed v necessary to build the barriers fast enough to actually contain the fire.

For specific results, Bressan initially focused on the following special case: The fire starts with the unit ball and its spread function is also governed by the unit ball, which means that it spreads at constant speed 1 uniformly in all directions. By rescaling arguments, all variants with the fire spreading uniformly in all directions can be normalized to this special case independent of the starting area. He further showed, that a building speed $v > 1$ is necessary and provided a strategy that shows that $v > 2$ is sufficient for this case, which he also conjectured to be optimal. Though both these initial

results are rather straight-forward, that gap of $(1,2]$ has remained open, despite a 500 USD reward offered by Bressan himself in 2011 [15].

Later research by Bressan et al. [14, 18, 19, 81] focused mainly on finding conditions for barrier arcs that are part of optimal strategies, in the sense that they minimize the area burned. For this, Bressan developed the notion of a *time efficiency* of barrier arcs, measuring how the time cost spent on building some barrier arcs is returned, when the fire is delayed and reaches other barriers later, allowing them to be built later as well.

It seems that the difficulty lies with the following question: *To contain a fire, should one build an enclosing barrier at maximum speed, or is it better to invest some time in building extra delaying barriers that will not be part of the final enclosure but can slow the fire down during construction?* If delaying barriers could be shown to be useless, Bressan's proof of the lower bound of 1 could be naturally extended to prove his conjecture, the lower bound of 2. In fact, Bressan et al. [17] also consider a special variant in where the fire spreads in a half plane. For this variant, they constructed a strategy without delaying barriers requiring building speed $v = 1$, that encloses the fire between the boundary of the half plane and the barrier curve and proved it to be optimal.

1.3.1 Related Problems

A special case of the firefighting problem, where only a single barrier curve is built in one continuous motion has also been considered by Bressan et al. in 2008 [21] providing a strategy, for which a building speed $v > 2.6144$ is sufficient. A slightly different strategy requiring the same speed was later discussed by Klein et al. [57] and shown to be optimal regarding the building speed v among a subset of spiralling strategies.

As in the discrete case, the continuous fire model also works well as a model for other phenomena featuring a some spreading contagion. The best fitting among these is probably an oil spill spreading on the surface of the ocean. The typical method to fight such spills is to deploy swimming containment booms by ship, which is well represented by the barrier curves in the plane. More generally, the modelling technique Bressan uses - defining the area burned by the fire as the reachable set of a differential inclusion of a set-valued function - has also been applied to other problems like herding flock [20] or modelling spreading disease [6].

Klein et al. [55] also studied controlling the spread of fire in restricted domains, namely polygons. As the fire is already contained in the bounded region of the polygon, the objective shifts instead to save as much of the polygon area as possible. They proved this problem to be NP-hard even when restricting the barriers to the diagonals of the polygon, but provided an approximation scheme as long as the barriers must come from a set of straight-line segments with both endpoints on the boundary of the polygon.

Kostitsyna et al. [60] also studied a related problem in polygons - given an entry and an exit interval on the boundary, they study how to build barriers to maximize the length of the shortest path

from entry to exit or minimize the biggest flow. This is similar to delaying a fire entering the polygon at the entry from reaching the exit as long as possible.

1.3.2 Our Contribution

In this thesis, we focus specifically on the aspect of delaying barriers, as understanding their value seems to be crucial to close the gap between the known lower and upper bound on the necessary building speed.

The central results are obtained for a specific scenario, where a fire spreading at unit speed according to the L_1-metric is to be contained in an open half-plane. To study the effect of delaying barriers, we focus on barrier systems that are a combination of the horizontal enclosing barrier along the boundary of the half-plane, as well as vertical delaying barriers attached to that boundary.

Intuitively, the fire can be contained in the upper half-plane with just the horizontal barrier and a building speed $v > 2$. Although one might initially expect this to be, in fact, optimal, we instead construct strategies that contains the fire for lower speeds. The best of those contains the fire for any $v > 1.8772$. To complement this constructive upper bound on the necessary building speed v, we also develop a series of lower bounds by careful analysis of the manner in which the fire buns along barrier systems of this specific structure, the strongest of which shows $v > 1.\overline{6}$ is necessary.

To provide additional context about the usefulness of delaying barriers in addition to these main results, we also examine three additional variants of geometric firefighting, for which the best known strategies involve delaying barriers to varying degrees. When restricting the firefighter by only allowing a single continuous barrier curve, the best known strategies (compare [21, 57]) involve spiralling around the fire origin, such that most of the barrier curve creates a single long delaying barrier. When restricting the fire, such that it only spreads along the rays from the origin, we present an optimal strategy for any positive speed that does not involve delaying barriers at all. Finally, we show that a variant of the half-plane model, with the fire restricted to only burning along barriers already built, allows for simple symmetric strategies with infinitely many delaying barriers.

1.4 The Minimum Enclosing Ball Problem

The *Minimum Enclosing Ball problem* (MEB) or Euclidean 1-Center problem is a well-established geometric clustering problem.

Minimum Enclosing Ball (MEB)
Input: *A set of points $P \subset \mathbb{R}^d$.*
Goal: *Find a ball containing P with minimum radius. This is equivalent to finding the center $c \in \mathbb{R}^d$ minimizing $\max_{p \in P} d(c,p)$, where $d(c,p)$ denotes the Euclidean distance between the points c and p.*

The Minimum Enclosing Ball problem relates to firefighting in two ways. First, consider a number of points in $\mathbb{R}^d$ representing sites that need to be protected from a fire using an $\mathbb{R}^{d-1}$-dimensional

barrier. One natural way to do so, would be to build the surface of a minimum enclosing ball, which needs to be found first. Second, the center of a minimum enclosing ball is the optimal position to build a firefighter station that minimizes the maximum response time of reaching a fire breaking out at any of the points.

For any constant dimension, the Minimum Enclosing Ball problem can be solved in linear time as shown by Megiddo in 1983 [64] by using a rather complicated prune and search approach. Based on the simple observation, that the minimum closing ball in a constant dimension d can be determined by at most $d+1$ of the points from P lying on its boundary, and a linear programming algorithm by Seidel [75], Welzl described a surprisingly simple randomized algorithm running in expected time $O(n)$ in 1991 [82]. However, both Megiddos and Welzls algorithm run in time $O((d+1)(d+1)!n)$ for general dimension d, although the latter provided experimental results of its efficiency in practice.

An algorithm by Gärtner et al. [40] based on quadratic instead of linear programming intended for dimensions up to 30 runs in practice in time polynomial in d. However, it requires arbitrary precision linear algebra to stay robust, which limits its use in higher dimensions. This problem is avoided by a purely combinatorial algorithm by Fischer et al. [35], which is another example of an exact algorithm with exponential runtime in the worst case, but good performance in practice up to roughly $d = 10000$. Until today, no exact algorithm with running time polynomial in d is known. Therefore, approximation algorithms have been considered.

Bădoiu et al. [9] presented an $(1+\varepsilon)$ approximation algorithm running in time $O(\frac{nd}{\varepsilon^2} + \frac{1}{\varepsilon^{10}} \log \frac{1}{\varepsilon})$. It is based on finding a core-set of size at most $\frac{1}{\varepsilon^2}$ - a subset of points $S \subset P$, such that the optimal MEB solution for S is an $(1+\varepsilon)$ approximation for the whole set P. Kumar et al. [61] improved these results to finding ε-core-sets of size $O(\frac{1}{\varepsilon})$ in time $O(\frac{nd}{\varepsilon^2} + \frac{1}{\varepsilon^{4.5}} \log \frac{1}{\varepsilon})$. Independently Bădoiu and Clarkson [7] achieved an algorithm with a slightly worse running time of $O(\frac{nd}{\varepsilon} + \frac{1}{\varepsilon^5})$ while having a stricter bound of $\lceil \frac{2}{\varepsilon} \rceil$ on the size of their core-set.

Bădoiu and Clarkson [7] also gave an incredibly simple gradient-descent algorithm obtaining a $(1+\varepsilon)$-approximation in time $O(\frac{nd}{\varepsilon^2})$ and later showed that a tight bound of $\lceil \frac{1}{\varepsilon} \rceil$ on the size of ε-core-sets exists, see [8].

Rösner [74] recently applied known results about optimization over convex bodies (see [43]) to obtain a $(1+\varepsilon)$-approximation algorithm in time and space polynomial to the encoding size of the problem instance and $\log \frac{1}{\varepsilon}$. For instances with rational or integral coefficients, his algorithm can even be used to obtain optimal solutions.

The core-set based algorithms, Bădoiu's gradient-descent algorithm as well as Rösner's convex optimization based algorithm can also be applied to the Euclidean k-center problem, a classic clustering problem and natural extension of the minimum enclosing ball problem:

Euclidean k-center
Input: *A set of points $P \subset \mathbb{R}^d$ and a value $k \in \mathbb{N}$.*
Goal: *Find k centers $c_1, \ldots c_k \in R^d$ minimizing $\max_{p \in P} \min_i = 1^k d(c_i, p)$, where $d(c_i, p)$ denotes the Euclidean distance between the center c_i and the point p.*

As shown by Megiddo [65, 66], the Euclidean k-center problem is NP-hard for $d \geq 2$ and does not have a polynomial time approximation scheme in high dimensions unless $P = NP$. For $d = 1$ however, it can be solved exactly in time $O(n \log n)$ [67]. For $d = 2$, an initial algorithm by Drezner [29] computing an optimal solution in time $O\left(n^{2k+1} \log n\right)$ was improved by Hwang et al. [45] to a runtime of $O\left(n^{\sqrt{k}}\right)$.

The case $d = 2$ and $k = 2$ has received particularly extensive attention resulting in a series of publications improving the running time step by step culminating in a randomized algorithm with expected running time $O\left(n^2 \log n\right)$ by Eppstein [31], which could only recently be matched by a deterministic algorithm by Tan and Jiang [77].

When restricted to rational instances $P \subseteq \mathbb{Q}^d$ with $|P| = n$, Rösner [74] gave an algorithm that computes an optimal Euclidean k-center solution in $O\left(n^{(d+1)k} \operatorname{poly}(\text{enc})\right)$ time, where enc is the length of the input encoding. It first computes a set of $O\left(n^{dk}\right)$ partitionings of the point set, which includes the optimal partitioning into clusters, and then tests all those partitionings by applying the 1-center algorithm to each partition.

The core-set based approximation algorithms can be extended to the k-center problem using a guessing oracle approach [9]. Assuming we had an oracle telling us for each point added to the core-set, to which of the k-clusters it belongs, it suffices to build a core-set with k times as many point as for the 1-center problem. Missing such an oracle, just exhausting all guesses results in a run time of $O\left(nd\, k^{\frac{k}{\varepsilon}}\right)$. As the running time is then exponential in the size of the core-set, a tight bound on that size is paramount for this approach. The same approach can be applied to the gradient-descent algorithm, in which case the number of points used for the gradient descent steps has the same effect on the run time as the size of the core-set.

1.4.1 Our Contribution

We extend a geometric property first used for a 2-center streaming algorithms by Kim and Ahn [47] to improve the gradient-descent approach and provide a $(1+\varepsilon)$-approximation with a run time of $O\left(\frac{nd}{\varepsilon}\right)$ using $\lfloor \frac{2}{\varepsilon} \rfloor$ many points. While this is an improvement on the known core-set-based algorithms only for $nd \in \Omega\left(\frac{1}{\varepsilon^{3.5}} \log \frac{1}{\varepsilon}\right)$ and the exact dependency on the input encoding size in comparison to the most recent result by Rösner is unclear, the simplicity of the algorithm might nevertheless make it an attractive option in real-world applications.

In clustering and Big Data, a simpler algorithm can be preferable to a more complex algorithm despite a worse worst-case runtime, as that simplicity usually results in low run time constants and easier adaptability. In the case of the 1-center problem, this is evident by the fact that Bădoiu and Clarkson considered an incredibly simple gradient-descent algorithm despite having found a core-set based algorithm of better asymptotic worst-case performance. They used performance experiments on both the gradient-descent and the different core-set-based algorithms to show that the gradient-descent algorithm converges significantly faster than its theoretical bound suggests [8].

1.5 Structure of this Thesis and Bibliographical Notes

After this introduction, the main body of this thesis consists of three chapters, each focusing on one of the three problems introduced. The results will be summarised in a final conclusion chapter.

Some results in this thesis have been presented at workshops or conferences, published in the respective proceedings or as journal versions. We provide an overview of previous publications here and also specify in which way this thesis extends or improves on these results.

Chapter 2 This chapter covers our contributions to discrete firefighting as detailed in Section 1.2. Preliminary versions of the results in this chapter have been presented at the *Conference on Algorithms and Discrete Applied Mathematics (CALDAM 2020)* and published in its proceedings [58] together with co-authors Rolf Klein, David Kübel, Elmar Langetepe and Jörg-Rüdiger Sack. A full version of the conference paper is available as an arXiv preprint [56] including some proofs omitted from the conference version due to space constraints. Some of the results focusing on the computability aspects of the new model are presented in the PhD-Thesis of David Kübel [46].

In contrast to these publications, this thesis generalizes both the model and the algorithmic results presented with respect to the underlying graphs. In addition, we explore the adaptability of our new model to several environmental factors affecting wildfires, as well as propose a number of open problems focused on the unique features of our model.

Chapter 3 This chapter covers our contributions to continuous firefighting as detailed in Section 1.3 The central results, with respect to containing a fire in the upper half-plane, have been presented at the *Algorithms and Data Structures Symposium (WADS) 2019* and published in its proceedings [49] with co-authors Elmar Langetepe, Sang-Sub Kim, Rolf Klein and David Kübel. A full version of the conference paper including some proofs omitted due to space constraints is available as an arXiv preprint [50] and has been submitted to *Computational Geometry: Theory and Applications*. Initial results had also been presented at the 34^{th} *European Workshop on Computational Geometry (EuroCG 2018)* [54] based on an extended abstract without formal publication.

As one of the problems presented to give context to the focus on delaying barriers we also summarize results from a previous publication with co-authors Rolf Klein, Elmar Langetepe, Christos Levcopoulos and Andrzej Lingas about a variant of geometric firefighting, in which the fire is contained by a single continuous barrier curve [57]. An earlier version of those results by Rolf Klein, Elmar Langetepe and Cristos Levcopoulos was presented at the 31^{st} *Symposium on Computational Geometry (SoCG 2015)* and included in its proceedings [53].

This thesis contains some unpublished material in addition to the results included in these publications: An optimal strategy to contain a fire spreading along rays from an origin point, and a strategy to prevent a fire that is only allowed to burn along existing barriers from entering the lower half-plane, are included to provide context about the importance of delaying barriers in different variations of geometric firefighting. A lower bound for strategies to contain a fire, that starts at some

small distance from the x-axis, in the upper half-plane, two additional lower bounds for the half-plane model based on different approaches than the main result, and a discussion how to extend the obtained results to higher dimensions or to a fire spreading according to the L_2-metric, provide additional insight into the central half-plane model.

Chapter 4 This chapter covers our contributions to the minimum enclosing ball problem as detailed in Section 1.4. The results in this chapter have been presented at the 36^{th} *European Workshop on Computational Geometry (EuroCG 2020)* [48] based on an extended abstract with co-author Sang-Sub Kim, but have not been formally published. This thesis completes that extended abstract with full proofs.

Chapter 2

A New Model for Discrete Firefighting

In this chapter we introduce a discrete model for a fire spreading through a graph, that is a direct extension of the model used in Hartnell's Firefighter Problem. We begin with the definition of the model and compare it to both the original model used by Hartnell and the related concept of cellular automata. While the model is defined on general graphs, it makes the most sense in simulating terrain when applied to graphs induced by a regular tiling of the plane.

We discuss a variety of problems based on the new model, mainly focusing on the graph induced by the regular hexagonal tiling of the plane. First we present an efficient algorithm to track the fire propagation that is particularly effective in graphs with bounded maximum vertex degree. Second we present algorithms for finding a barrier to protect a specific set of target vertices representing a village in graphs induced by a hexagonal tiling of the plane. At last we prove that the problem of choosing to ignite a minimum subset of some candidate starter vertices to eventually ignite a set of target vertices is NP-complete, even in rectangular grid graphs.

As flexibility was the focus of the creation of this new model, we then discuss a few ways to extend or adapt this model to reflect multiple environmental aspects affecting real world fire propagation. We also propose a number of problems that could be studied utilizing the unique properties of the new model.

2.1 Model Definition

The goal of our model is to better reflect the properties of terrain affecting the spread of a fire by extending the simple model of a burning vertex igniting all neighbours.

On the one hand, depending on varying parameters (type and thickness of vegetation, humidity and more) an area of terrain might be harder or easier to ignite by a spreading fire. For example, a dry wheat field is much easier to ignite than a thick forest full of fresh moist greenery. We model this as the resistance r of a vertex, which is reduced by burning neighbours until a vertex ignites when its resistance reaches 0.

On the other hand, terrain does not burn indefinitely after catching fire, but different terrain might burn for varied durations and with varied intensity, which makes it easier or harder for the fire to further spread to the surrounding area. While the dry wheat field burns out fast, the thick forest will burn much longer and have a higher chance of spreading the fire further. We model this as the energy e of a vertex, which decreases over time until a vertex burns out and no longer affects its neighbours when its energy reaches 0.

Definition: Resistance and Energy Model (r,e-Model)
Let $G=(V,E)$ be a connected graph, and $r_0 : V \mapsto \mathbb{N}$, $e_0 : V \mapsto \mathbb{N}$ two initialization functions assigning a starting resistance *and* energy *to each vertex v in the graph. We call such a triplet (G,r_0,e_0) an instance of the r,e-model. Then the state of a cell at a discrete time $t \in \mathbb{N}$ is given by two attributes: its energy $e(v,t)$ and resistance $r(v,t)$ with $e(v,0)=e_0(v)$ and $r(v,0)=r_0(v)$. A vertex v is called*

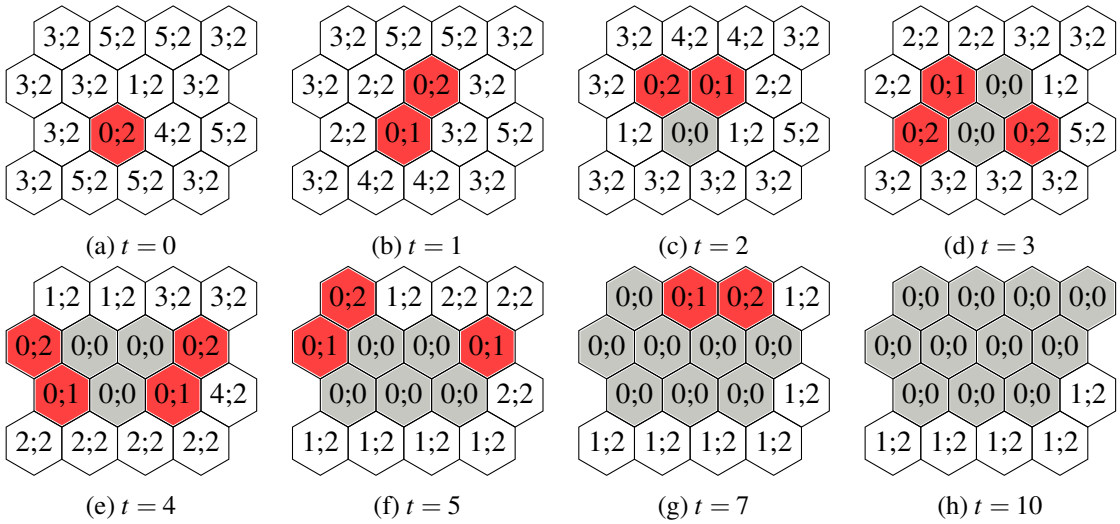
- alive *at time t if $r(v,t)>0$,*
- burning *if $e(v,t)>0$ and $r(v,t)=0$ and*
- dead *if $e(v,t)=0$ and $r(v,t)=0$.*

The following transition rules for a vertex v model the spread of the fire through the graph:

- *If v is* alive *at time t, then $e(v,t+1) \coloneqq e(v,t)$ and $r(v,t+1) \coloneqq \max\{r(v,t)-b,0\}$, where b denotes the number of direct neighbours of v in G burning at time t.*
- *If v is* burning *at time t, then $r(v,t+1) \coloneqq r(v,t)=0$ and $e(v,t+1) \coloneqq e(v,t)-1$.*
- *If v is* dead *at time t, then $r(v,t+1) \coloneqq r(v,t)=0$ and $e(v,t+1) \coloneqq e(v,t)=0$.*

When using the r,e-model to represent a fire spreading through a terrain, it is intuitive to consider graphs induced by tessellations. A tessellation is a non-overlapping tiling of the plane with polygons. A tessellation is called *regular*, if the tiles are all congruent regular polygons. The only regular polygons, for which a regular tessellation exists are triangles, squares and hexagons.

A tessellation T induces an infinite planar graph $G_T=(V_T,E_T)$: each polygon in T corresponds to a vertex v in V_T and E_T contains an edge (u,v) if and only if the polygons corresponding to u and v share a polygon edge in the tessellation T. Due to its regularity, we mainly focus on the graph G_{hex} induced by the regular hexagonal tiling (and finite subgraphs of it) and refer to its tiles and the corresponding vertices as cells. Figure 2.1 shows an example of how a fire expands over time from a single source in a small subgraph of 4×4 hexagonal cells. That example also highlights, that there are instances in which the fire burns out on its own, without igniting every vertex of the graph G.



(a) $t = 0$ (b) $t = 1$ (c) $t = 2$ (d) $t = 3$

(e) $t = 4$ (f) $t = 5$ (g) $t = 7$ (h) $t = 10$

Figure 2.1: Fire spreading in a finite subgraph of G_{hex}. The resistance and energy are given in the cells as pairs of the form r;e. The state of the cells are further indicated by colours: Burning cells are red, alive cells are white, and dead cells are grey. At time $t = 10$, the fire has burned out and several living cells remain.

2.2 Related Models

2.2.1 Hartnell's Firefighter

In the original Firefighter problem as introduced by Hartnell in [44] (compare Section 1.2), a burning vertex in a graph ignites each non-burning adjacent vertex each round. As intended, the r,e-model generalizes this fire propagation model:

Let (G,s) be an instance of Hartnell's firefighter problem. Then consider an instance of the r,e-model of the same graph G with $e_0(v) = 1$ for all vertices and $r_0(v) = 1$ for all vertices except the starting vertex s, for which $r_0(s) = 0$. Thus, the information about the starting vertex is implicitly included in r_0. With that initialization, any vertex ignites all non-burning neighbours within one time step. This makes the set of all burning and dead vertices at time t in that r,e-model instance the same as the set of vertices on fire in round t in Hartnell's model, while the set of the burning vertices in the r,e-model at time t is exactly the vertices ignited in round t in Hartnell's model.

Consequently, Hartnell's firefighter problem (and other problems using the same propagation model) can be adapted to this model:

r,e-Firefighter

Input: *An instance of the r,e-model.*

Consider the following game played in rounds. Each round, a firefighter is allowed to mark a non-burning vertex as protected. Then the fire affects non-protected vertices according to the transition rules of the r,e-model. The game is played until there are no more non-protected vertices adjacent to a burning vertex.

Goal: *Find the sequence of vertices to protect that minimizes the number of burning or dead vertices at the end of the game.*

The construction described above is a polynomial time reduction of Hartnell's Firefighter problem to the r,e-Firefighter Problem. Thus, the r,e-Firefighter Problem is NP-complete for graph classes, for which Hartnell's Firefighter problem is NP-complete, like bipartite graphs [63], trees of maximum degree three [34], and cubic graphs [51].

On the other hand, the fire according to the r,e-model burns at most as fast through a graph G as Hartnell's fire.

Lemma 1
Let $G=(V,E)$ be a graph with initialization functions $r_0 : V \mapsto \mathbb{N}$ and $e_0 : V \mapsto \mathbb{N}$, such that there is a unique starting vertex $s \in V$ with $r_0(s) = 0$. Let $F_{re}(t)$ be the set of burning or dead vertices according to the r,e-model at time t and $F_H(t)$ be the set of cells on fire after round t of fire spreading from s according to Hartnell's model. Then $F_{re}(t) \subseteq F_H(t)$.

Proof. The claim follows by induction. As a start, $F_{re}(0) = \{s\} = F_H(0)$. Now assume, $F_{re}(t) \subseteq F_H(t)$ for some fixed t. Then in Hartnell's model, every neighbour of a burning vertex gets ignited. In the r,e-model, a vertex ignites, when its resistance is reduced to 0. The resistance of a vertex is only reduced in a round if it has burning neighbours, so the ignited vertices are a subset of all neighbours of burning vertices, which again is a subset of $F_{re}(t)$ which concludes the proof. □

By Lemma 1, any strategy for Hartnell's Firefighter model saves at least as many vertices when applied to an instance of the r,e-Firefighter Problem with a unique starting vertex s.

Similar relations can be found when adapting the results for the budgeted Firefighter problem on regular grid graphs as introduced in Section 1.2. As a reminder, in the budgeted variant of the Firefighter problem, the fighter is allowed to protect not only 1 but b vertices each round according to a budget b. In infinite graphs, the game only ends if the fire is contained, i. e. when there is no non-protected alive vertex adjacent to a burning vertices. Therefore, the budgeted Firefighter problems asks two questions. First, what is the minimum budget b to contain the fire in some regular grid graph. Second, given a budget b, give an optimal strategy that minimized the number of burned vertices.

By Lemma 1, if a budget b suffices to contain a fire spreading from a single vertex in some n-dimensional regular grid-graph G, then the same budget suffices for every instance of the r,e-Firefighter problem on that graph with a unique starting vertex s. However, depending on the initial resistance and energy values, the fire might burn out on its own without any interference from a firefighter. Therefore, for any given instance of the r,e-Firefighter problem on G, the optimal firefighter budget to contain the fire lies somewhere between 0 and b.

2.2.2 Cellular Automata

A cellular automaton is defined on cells in a regular tiling, typically (but not limited to) the square tiling. Each cell is assigned a starting state at time $t = 0$ from a finite set of states. In each subsequent discrete

time step (i. e. $t \in \mathbb{N}_{>0}$), the state of cell is determined by the states of the cell in its neighbourhood in the previous time step based on a fixed rule set. A general introduction can be found in [79].

When applied to a regular grid of square cells, i. e. a graph induced by a regular tessellation, the r,e-model defines a cellular automaton as long as the initial resistance and energy are bounded by a constant c. Although the set of states seems to be infinite at first glance, observe that both energy and resistance of a cell never increase. Thus, the finite set of states is the set of pairs (r,e) with $r,e \in \{0,\ldots,c\}$. Note, that the fire propagation model used in Hartnell's Firefighter Problem also defines a cellular automaton, albeit a very simple one with just two states and a single rule, allowing a one-time transition from alive to burning.

Originally introduced as "cellular spaces" by von Neumann and Ulam to model biological self-replication [80], it has been shown, that there exist computation-universal cellular automata, i. e. some cellular automata can simulate a universal Turing machine [76]. Among them (as shown by Rendell [73]) is the probably most famous cellular automaton: Conway's "Game of Life", which first appeared in Martin Gardner's column "Mathematical Games" [38]. It plays out on the orthogonal grid of square cells and each cell lives or dies dependent on its eight neighbours. A living cell survives to the next time step, if and only if it has two or three live neighbours. A dead cell on the other hand becomes a living cell, if and only if it has exactly three live neighbours. These rules are inspired by aspects affecting the survival of real life populations, like over- and underpopulation as well as reproduction.

What contrasts the r,e-model to typical cellular automata like Conway's "Game of Life" is it's monotonicity. Both the resistance r and the energy e of a cell will only decrease, never increase. Thus, a dead cell with state $(0,0)$ will stay in that state forever and the number of times a single cell v will change its state is limited by $r_0(v)+e_0(v)$. Therefore, if the automaton is limited to a finite number of cells, it will always eventually arrive at a still state with all cells either alive or dead, but not burning. In contrast, cells in Conways "Game of Life" can repeatedly switch between dead and alive, and infinitely oscillating structures are known on bounded grids of sizes as small as 3×3 cells.

This variety in behaviour and power of cellular automata makes them notoriously hard to rigorously classify. The most notable classification of cellular automata was introduced by Wolfram in "A new kind of Science" [83], and its 4 classes are not based on formal definitions, but different types of behaviours Wolfram observed in a huge variety of 1-dimensional cellular automata. Essentially, Wolfram splits cellular automata into those that always quickly evolve into (relative) stability or patterns oscillating within very few rounds (class 1 and 2), pure chaos (class 3) or complex interacting structures surviving for many rounds (class 4).

This complex behaviour of the last class, to which Conway's "Game of Life" belongs, seems to enable universal computation. As such, Wolfram conjectured that most, if not all, cellular automata belonging to that class have the same capability of universal computation. It is clear, that some initial patterns in our r,e-model can enter a stable state very quickly if the fire burns out fast, fitting the class 1 or 2. But the fire might also burn on forever without ever stabilizing or oscillating, as it does

when initializing all cells with r_0 and e_0 set to 1 except for a single cell s with $r_0(s) = 0$ instead, which suggests class 4 instead. Due to the inherent fuzziness of the classification, our model can not necessarily be classified with certainty.

However, we will later shortly address a variation of the r,e-model in Section 2.5, that is capable of simulating Turing machines as proven by David Kübel in his PhD-Thesis [46]. However, this computational power either requires additional transformation rules for our r,e-model, or an underlying graph different from the 2-dimensional grids of cellular automata.

2.3 An Efficient Algorithm for Fire Propagation

When studying firefighting strategies, we need a baseline to compare any firefighting effort to, i. e., what happens without any interference. Therefore, given an instance (G, r_0, e_0) of the r,e-model, one might be interested in the situation at a specific time, i. e., the state of every vertex $v \in V$ at time step t. We call this the *snapshot* of an instance (G, r_0, e_0) at time t.

In the similar model for Hartnell's Firefighter Problem, this question is easy to answer for an instance (G,s). As the state of a vertex is binary (burning or alive), finding the snapshot at time t reduces to finding the set of vertices ignited in rounds 1 to t. As the fire spreads to all adjacent vertices each round, the vertices ignited in round 1 are exactly the vertices with distance 1 to s . In the next round, all non-burning neighbours of these vertices are ignited, which are exactly the vertices at distance 2 to s. Inductively, the vertices with a distance k to s are ignited in round k. As G is an undirected unweighted graph, a simple Breadth-First-Search algorithm running in time $O(|E|+|V|)$ is enough to determine the ignition round of every vertex, which allows to return the snapshot for any round t in the output time $O(|V|)$.

It is not as simple in the r,e-model, even if there is only a unique starting vertex s with $r_0(s) = 0$ as in Hartnell's model. The naive way to calculate a snapshot is to start with the snapshot at time $t = 0$ as given directly by the instance and do a step-by-step simulation. This simulation terminates either when t is reached or when the fire burns out and no vertex changes its state from one round to the next. This can take as much as $|V| \cdot e_{max}$ many rounds, where $e_{max} \coloneqq \max_{v \in V} e_0(v)$ is the highest starting energy of any vertex. Each round, $O(|E|)$ edges between alive and burning vertices can affect the state of vertices in the next round. Together, this yields the following theorem:

Theorem 1
Using a step-by-step simulation for an instance of the r,e-model, a snapshot at time t can be calculated in time $O(|E| \min\{t, |V| \cdot e_{max}\})$.

Instead of this naive approach, one can use a similar approach as using *BFS* in Hartnell's Model to determine the ignition round of each vertex and use this as a preprocessing step. Given an instance (G, r_0, e_0) of the r,e-model, let $t_i(v)$ be the *ignition time* of a vertex v, i. e. the smallest time t, for which it holds $r(v,t) = 0$, or ∞ if no such t exists. We will present an algorithm to compute $t_i(v)$ for

all vertices $v \in V$, as well as show how to use this preprocessing step to compute a snapshot at any t, culminating in the following theorem:

Theorem 2
For an instance (G, r_0, e_0) of the r,e-model, the ignition time of every vertex in G can be computed in time $O(|E| \cdot |V|)$. A snapshot at time t can be computed based on the ignition times in additional time $O(|V|^2)$. For graphs with maximum vertex degree bounded by a constant, these runtimes can be reduced to $O(|V| \log |V|)$ for the ignition times and an additional $O(|V|)$ for the snapshot.

Crucially, the runtime of both steps is only dependent on the size of the graph G and neither on the initial resistance and energy values given by e_0 and r_0, nor the time step t. As long as $t \in \Omega(|V|)$, this approach is better than the naive step-by-step simulation, especially if we want to compute multiple snapshots. For graphs with maximum vertex degree bounded by a constant, like graphs induced by regular tessellations, the difference becomes even stronger. In this case $|E| \in O(|V|)$, so the runtime of the step-by-step algorithm reduces to $O(\min\{t, |V| e_{max}\}|V|)$. This means the preprocessing approach is better for $t \in \Omega(\log |V|)$.

2.3.1 An Efficient Algorithm to Compute Ignition Time

Let $\mathcal{N}_v$ be the neighbourhood of a vertex v, i. e. $\mathcal{N}_v := \{u \in V \mid \{u, v\} \in E\}$. By the propagation rules of the r,e-model, these are the only vertices that affect $t_\mathrm{i}(v)$.

Let n be an arbitrary neighbour of v. Until time $t_i(n)$, n is alive and hence does not affect $r(v)$. After n ignites at round $t_i(n)$, it burns, reducing the resistance of v by one each round, until either v ignites, or n dies after $e_0(n)$ rounds. At an arbitrary time t, n can therefore have reduced the resistance of v by at most $\max\{0, \min\{t - t_\mathrm{i}(n_i), e_0(n_i)\}\}$.

This gives us a formal definition of the ignition time:

$$t_\mathrm{i}(v) = \min\left\{\left\{t \in \mathbb{N} \;\middle|\; r_0(v) \leq \sum_{n \in \mathcal{N}_c} \max\{0, \min\{t - t_\mathrm{i}(n), e_0(n)\}\}\right\} \cup \{\infty\}\right\}.$$

Now, let $N \subseteq \mathcal{N}_v$ be a subset of the neighbours of v for which t_i is known. We define the *partial ignition time* $t_\mathrm{pi}(N, v)$ to be the time at which v would ignite if the vertices in N were the only neighbours of v, so

$$t_\mathrm{pi}(v) := \min\left\{\left\{t \in \mathbb{N} \;\middle|\; r_0(v) \leq \sum_{n \in N} \max\{0, \min\{t - t_\mathrm{i}(n), e_0(n)\}\}\right\} \cup \{\infty\}\right\}.$$

As we are working with a subset of the actual neighbours of v, it is clear, that $t_\mathrm{pi}(N, v) \geq t_\mathrm{i}(v)$ for any $N \subseteq \mathcal{N}_v$. Our algorithm is mostly based on the following intuitive observation: A neighbour n of v that ignites after v does not actually affect $t_\mathrm{i}(v)$, which corresponds to the following formal Lemma.

Lemma 2

Let v be a cell and $\mathcal{N}_v$ its neighbourhood. If $N = \{n \in \mathcal{N}_v \mid t_i(n) < t_i(v)\}$, then $t_{pi}(N, v) = t_i(v)$ holds.

Proof. Obviously, if $N = \mathcal{N}_v$, then $t_{\mathrm{pi}}(N, v) = t_{\mathrm{i}}(v)$. Consider a vertex $n_i \in \mathcal{N}_v$ that is not in N; hence $t_{\mathrm{i}}(n_i) \geq t_{\mathrm{i}}(v) \Leftrightarrow t_{\mathrm{i}}(v) - t_{\mathrm{i}}(n_i) \leq 0$ holds and consequently $\max\{0, \min\{t - t_{\mathrm{i}}(n_i), e_0(n_i)\}\} = 0$. This implies that n_i does not affect $t_{\mathrm{i}}(v)$:

$$\begin{aligned} t_{\mathrm{i}}(v) &= \min\left\{\left\{t \in \mathbb{N} \;\middle|\; r_0(v) \leq \sum_{n \in \mathcal{N}_v} \max\{0, \min\{t - t_{\mathrm{i}}(n), e_0(n)\}\}\right\} \cup \{\infty\}\right\} \\ &= \min\left\{\left\{t \in \mathbb{N} \;\middle|\; r_0(v) \leq \sum_{n \in \mathcal{N}_v \setminus \{n_i\}} \max\{0, \min\{t - t_{\mathrm{i}}(n), e_0(n)\}\}\right\} \cup \{\infty\}\right\} \\ &= t_{\mathrm{pi}}(\mathcal{N}_v \setminus \{n_i\}, v), \end{aligned}$$

which concludes the proof □

Using this observation, we can define an algorithm very similar to Dijsktra's shortest path algorithm to compute $t_{\mathrm{i}}(v)$ for all vertices v. The idea of Algorithm IGNITIONTIME is to compute t_{i} of all vertices in ascending order. Starting with the vertices s with $r_0(s) = 0$, which have $t_{\mathrm{i}}(s) = 0$, vertices are added to the a set C of 'correct' vertices in order of their partial ignition times computed based on the vertices in C.

Algorithm 1: IGNITIONTIME (G,r_0,e_0)

Input : An instance (G, r_0, e_0) of the r,e-model
Output : $t_{\mathrm{i}}(v)$ for all $v \in V$.

```
1  C = ∅
2  foreach vertex v ∈ V with r_0(v) > 0 do
3  |   t_i(v) ← ∞
4  foreach vertex v ∈ V with r_0(v) = 0 do
5  |   t_i(v) ← 0
6  while there is a vertex v ∉ C with t_i(v) < ∞ do
7  |   get v ∉ C with minimum t_i[v]
8  |   C ← C ∪ {v}
9  |   foreach neighbour n ∈ N_v with n ∉ C do
10 |   |   t_i(n) ← t_pi(N_n ∩ C, n)
11 return t_i(v) for each v ∈ V
```

2.3.2 Correctness of Algorithm IGNITIONTIME

Let $t'_i(v)$ be the ignition time as computed by our algorithm. To prove the correctness, we need to show that $t'_i(v) = t_i(v)$ for all vertices in V, when the algorithm terminates. We prove this separately for vertices in C and not in C at the termination of the algorithm.

Lemma 3
For each $v \in C$, $t'_i(v) = t_i(v)$.

Proof. We prove this by induction on $|C|$, as one vertex is added to C in each step. The first vertex added to C must be a vertex with $t'_i(v) = 0 = r_0(v) = t_i(v)$. Now let v be the last vertex added to C in an arbitrary step. As our induction hypotheses, it holds $t'_i(v') = t_i(v')$ for all $v' \in C \setminus \{v\}$. We need only show that $t'_i(v) = t_i(v)$ to conclude the proof.

It holds by construction $t'_i(v) = t_{\mathrm{pi}}(\mathbb{N}_v \cap C, v) \geq t_i(v)$. By Lemma 2 and the induction hypothesis, if all neighbours n of v with $t_i(n) < t_i(v)$ are in C, then $t'_i(v) = t_i(v)$. Assume there is at least one neighbour n of v with $t_i(n) < t_i(v)$ not in C. Then again by construction, it holds $t'_i(n) = t_{\mathrm{pi}}(\mathbb{N}_n \cap C, n) \geq t_i(n)$. Thus, if all neighbours n' of n are in C with $t_i(n') < t_i(n) < t_i(v)$, then $t'_i(n) = t_i(n) < t_i(n) \leq t'_i(v)$. As the algorithm added v to C, this is a contradiction as $t'_i(v)$ must be minimum among all vertices not in C. So there must be another neighbour n' of n with $t_i(n') < t_i(n) < t_i(v)$ not in C. We can repeat this argument $|V|$ many times showing that there must be more than $|V|$ many distinct vertices of strictly decreasing t_i not in C. This contradiction concludes the proof. □

Lemma 4
For each v from V with $v \notin C$, $t'_i(v) = t_i(v)$.

Proof. Consider an arbitrary vertex v from V with $v \notin C$ at the termination of our algorithm. Then $t'_i(v) = \infty$ and, by construction and Lemma 3, $t'_i(v) = t_{\mathrm{pi}}(\mathbb{N}_v \cap C, v) \geq t_i(v)$.

Now assume $t_i(v) < \infty$. Then by the same argument as in the previous proof, there must be a neighbour $n \notin C$ of v with $t_i(n) < t_i(v)$. The same argument can be repeated $|V|$ times, showing that there must be more than $|V|$ many vertices from V with distinct t_i not in C. This contradiction concludes the proof. □

Lemma 3 and Lemma 4 together prove the correctness of Algorithm IGNITIONTIME.

2.3.3 Runtime of Algorithm IGNITIONTIME

It remains the question of an efficient implementation of Algorithm IGNITIONTIME. We can use a priority queue to store the at most $|V|$ vertices not in C with their $t_i(v)$ as the priority key. Using strict Fibonacci heaps for the implementation, we can return and remove the vertex with minimum key $t_i(v)$ in time $O(\log|V|)$ and insert new items or decrease their key in constant time $O(1)$ [22]. As each vertex is removed from the queue at most once, this takes a total time $O(|V|\log|V|)$. Initialization and output both take $O(|V|)$.

The remaining cost is the computation of $t_{\mathrm{pi}}(\mathcal{N}_n \cap C, n)$ and the subsequent updates of the priority keys. Such a computation happens at most once for every edge, so $O(|E|)$ times. To determine this, we will first show how to efficiently compute $t_{\mathrm{pi}}(\mathcal{N}_v, v)$ for any vertex v.

Lemma 5

Let (G, r_0, e_0) be an instance of the r,e-model and v be a vertex in G of degree d. Then given $t_i(n)$ for each neighbour $n \in \mathcal{N}_v$, ignition time $t_i(v)$ can be computed in time $O(d \log d)$.

Proof. The time during which a neighbour n burns corresponds to an interval $[t_i(n), t_i(n) + e(n,0)]$, as illustrated in Figure 2.2. We use a sweep-algorithm over the start- and endpoints of these intervals to compute $t_i(v)$. We first need to order the start- and endpoints using time $O(d \log d)$.

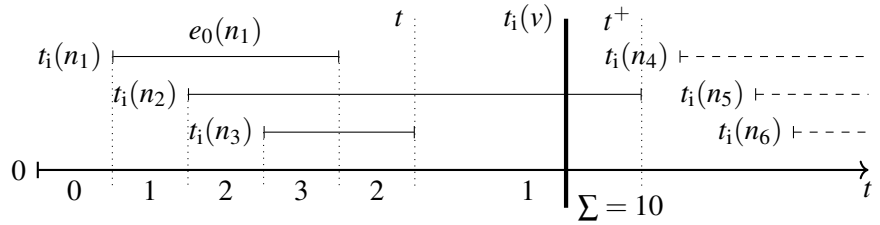


Figure 2.2: Each neighbour n_i of v starts burning at time $t_i(n_i)$ and burns for $e_0(n_i)$ rounds. The number of active intervals denoted below the axis is the amount, by which $r(v)$ is reduced each round. The dashed intervals belong to neighbours of v that ignite after v and thus do not affect $t_i(v)$.

Let the vertices $n_1, \ldots, n_d$ be the neighbours of v in ascending order of ignition time, i. e. $t_i(n_i) \leq t_i(n_2)$. During the sweep we maintain $r(v,t)$ and the number of burning neighbours b, i. e. the number of intervals intersected by the sweep line. We begin with $b = 1$ and $r(v, t_i(n_1)) = r(v,0)$ at the start of the first interval. At an arbitrary interval start- or endpoint t with b active intervals, we can determine $r(v,t^+)$ for the next interval-start or endpoint t^+ in constant time as $r(v,t^+) = \max\{0, r(v,t) - b(t^+ - t)\}$. If this is 0, then we can find our target $t_i(v)$ in constant time as $t^+ \geq t_i(v) = \lceil t + \frac{r(v,t')}{b'} \rceil$.

If we reach the last interval endpoint with $r(v,t) > 0$, then $t_i(v) = \infty$. As we have $O(d)$ many interval start- and endpoints and each can be handled in constant time, the runtime of this algorithm is dominated by the sorting step taking $O(d \log d)$. □

The same sweep-algorithm can be applied to the computation of $t_{pi}(\mathcal{N}_n \cap C, v)$ for a neighbour of a vertex v newly added to C. However, each time we compute $t_{pi}(\mathcal{N}_n \cap C, n)$, only a single vertex has been added to $\mathcal{N}_n \cap C$, namely v. So if we keep the sorted list of interval start- and endpoints between updates, we don't need to spend $O(d \log d)$ time sorting everything again. Instead, $O(\log d)$ is enough to insert the endpoints of the new interval for v. Therefore, every single computation of $t_{pi}(\mathcal{N}_n \cap C, v)$ can be done in time $O(d_{max})$, where d_{max} is maximum vertex degree of any vertex in V. There are at most $O(|E|)$ many such computations (one for each pair of adjacent vertices), so the total time spent on computing partial ignition times is $O(|E| d_{max}) \in O(|E||V|)$. As $|E| \in \Omega(|V|)$ for connected graphs, this dominates the queue handling costs of $O(|V| \log |V|)$ for retrieving v each round as well as the initialization costs of $O(|V|)$.

Thus, Algorithm IGNITIONTIME runs in time $O(|E||V|)$.

If the degree of every vertex in V is bounded by a constant, the sweep subroutine to compute t_{pi} runs in constant time. In addition $|E| \in O(|V|)$, which means the total time spent on computing

partial ignition times is only $O(|V|)$. Thus, the handling cost of the priority queue for retrieving v each round dominates the runtime of Algorithm IGNITIONTIME instead, which therefore runs in time $O(|V|\log|V|)$ in graphs of bounded maximum vertex degree.

2.3.4 Computing the Snapshot

Given t_i for all vertices in V, we can compute the state of a vertex v at time t with the same sweep approach as used to compute t_{pi} in the proof of Lemma 5. If $t > t_i(v)$, then $r(v,t) = 0$ and $e(v,t) = \max\{0, t - t_i\}$. Otherwise, $e(v,t) = e_0(v)$ and we can determine $r(v,t)$ based on the last interval endpoint before t. As we really only require the ignition time of all vertices v, that have $t_i(v) \leq t$, we can save a little time by terminating Algorithm IGNITIONTIME, as soon as the minimum ignition time of any vertex not in C is larger than t. If we remember, where we stopped Algorithm IGNITIONTIME, we can continue where we left of, if we want to compute an additional snapshot for some larger t.

When computing a snapshot, we need the state of every vertex v, so we can save some time again by globally sorting the interval end points once instead of doing it separately for every vertex. This reduces the runtime to $O\left(|V|^2 + (|V|\log|V|)\right)$. If the graph has bounded vertex degree, however, the sweep algorithm runs in constant time, which means we can compute the snapshot in time $O(|V|)$.

This concludes the proof of Theorem 2.

2.4 Results for Regular Tilings

There is a multitude of interesting questions one can formulate within this model. To choose a graph that nicely represents an area, we will focus on the graph G_{hex} induced by the regular hexagonal tiling of the Euclidean plane, and the graph G_{square} induced by the regular square tiling. As each vertex of these graphs corresponds to a hexagonal or square tile, we will also refer to them as cells.

More specifically, we will focus on specific subgraphs G of G_{hex}, fulfilling additional structural conditions. Clearly, by the regularity of the hexagonal tiling, every cell in G has at most 6 neighbours. We refer to the set cells with exactly 6 neighbours as the *interior* of G and to its members as *interior cells*. Equivalently, we refer to the set of cells with less than 6 neighbours as the *boundary* of G and to its members as *boundary cells*.

Definition: Smooth Hexagonal Graph
A smooth hexagonal graph *is a finite, connected subgraph of G_{hex}, such that*

i) the hexagonal tiles corresponding to the vertices in G cover a simply connected region of the Euclidean plane,
ii) and the boundary cells of G form a unique cycle, which will be referred to as the boundary cycle *of G.*

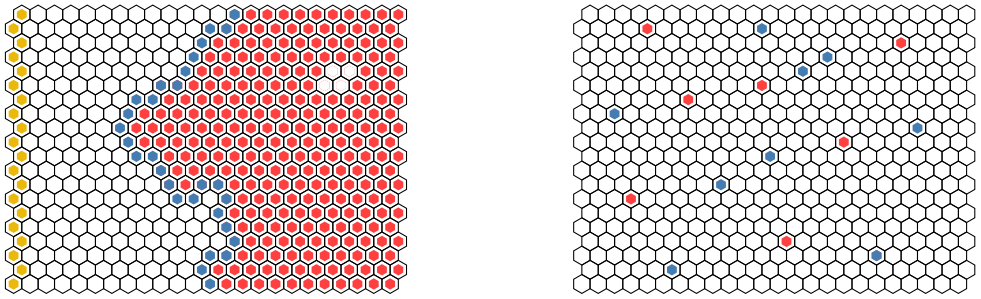
We equivalently define the class of Smooth Square Graphs for finite connected subgraphs of G_{square}. In this thesis, we will address three problems on smooth hexagonal graphs or smooth square graphs.

For the first two problems we will focus on a *village protection instance* (G, r_0, e_0, T). This is an instance of the r,e-model together with a set of cells T representing a village, that fulfils some additional structural properties.

i) G is a smooth hexagonal graph;
ii) the set $S \subset V$ of initially burning cells forms a subpath of the boundary cycle of G;
iii) and the non-empty set of village cells $T \subset V \setminus S$ forms a subpath of the boundary cycle of G.

Note, that while T is explicitly given in the instance, S is implicitly given by r_0, as the set of initially burning cells are exactly all cells c with $r_0(c) = 0$. But we require both S and T to form subpaths of the boundary cycle of G.

To protect the village cells from the fire, a firefighter can preventively fortify some of the cells not initially on fire by increasing their resistance r. In the first version of this problem discussed in Subsection 2.4.1, all cells from a protective set P have their resistance increased by the same amount k. This could correspond to a fly-over by aircraft that douses each cell with the same amount of water. The goal is to both find the minimum k, such that the village can be protected, and the minimum cardinality set P for this k.



(a) A protective set (blue cells) prevents a fire (red cells) from reaching the village (yellow cells).

(b) Given two set of cells S (red) and T (blue). Can m cells out of S set on fire burn down *all* cells of T?

Figure 2.3: Problem variants on a smooth hexagonal graph.

In the second version of this problem discussed in Subsection 2.4.2, firefighters can increase the resistance of cells individually. This might correspond to the firefighters thinning out parts of the forest and removing dry timber to make it harder for the fire to spread. Now we are interested in finding a protective set P for which the sum of these resistance increments of cells in P is minimal; see Figure 2.3a.

The third problem is an inverse variant of these problems. Given an instance (G, r_0, e_0) the goal is to ignite a set of target cells $T \subset V$. To achieve this goal, one may set a number of cells from a set $S \subset V$ on fire, i. e. change r_0 to 0 for these cells; see Figure 2.3b. The optimization question is then

as follows: Are there m cells in S which, when set on fire, will eventually ignite all cells in T? In Subsection 2.4.3 we prove this problem to be NP-complete, even if G is a smooth square graph,.

2.4.1 The k-Protection Problem

Let (G, r_0, e_0, T) be a village protection instance. Assume that, before the fire starts to spread, some firefighters are allowed to protect the village cells by dousing some cells with water to increase their resistance by a constant amount k. We call a subset P of $V \setminus S$ a k-protective set, if increasing the resistance of every cell in P by k results in no cell from T igniting. More precisely, P is a k-protective set, if and only if $t_i(v) = \infty$ for all village cells $v \in T$ in a modified instance (G, r'_0, e_0), with $r'_0(c) = r_0(c) + k$ for all cells c in P, and $r'_0(c) = r_0(c)$ otherwise.

k-Protection
Input: *A village protection instance* (G, r_0, e_0, T).
Goal: *Find the minimum k, such that a k-protective set P exists.*

To solve the k-Protection Problem, we first study the corresponding decision problem: Given (G, r_0, e_0) and k, does a k-protective set P exist?

Clearly, if P is a k-protective set, then any superset P' of P with $P' \subset V \setminus S$ is a k-protective set as well. By definition, $V \setminus S$ is a superset of any k-protective set P. Thus, a k-protective set exists if and only if $V \setminus S$ is a k-protective set. We can test this applying Algorithm IGNITIONTIME presented in Section 2.3 to determine the ignition time of all village cells in the modified instance (G, r'_0, e_0). By Theorem 2, this takes $O(|V| \log |V|)$ time, as G has a bounded maximum vertex degree of 6.

Intuitively, if P is a k-protective set, then it also is a k'-protective set for any $k' > k$. Thus, we can combine our algorithm for the decision problem with binary search over possible k to solve the minimization problem. It remains to find a suitable upper bound for k. Let e_{maxSum} be the maximum sum of initial energy values of all direct neighbours of a cell, over all cells in the grid. A cell with more initial resistance than the sum of its neighbours initial energy survives even if all neighbours ignite. Hence, $V \setminus S$ is a e_{maxSum}-protective set, so $0 \leq k \leq e_{maxSum}$ holds, which concludes the proof of the following theorem.

Theorem 3
The k-Protection Problem can be solved in time $O(|V| \log |V| \log e_{maxSum})$.

Note that this approach trivially extends to arbitrary graphs and arbitrary, but disjoint, sets $S \subset V$ and $T \subset V$. The only difference is the increased runtime of Algorithm IGNITIONTIME for general graphs.

Minimum Cardinality k-Protection

The k-Protection Problem gives rise to another closely related optimization problem.

The Minimum Cardinality k-Protection Problem
Input: *A village protection instance (G, r_0, e_0, T), as well as a protection parameter k.*
Goal: *Find a k-protective set P of minimum cardinality.*

To solve the k-Protection Problem, we used Algorithm IGNITIONTIME to test whether $V \setminus S$ is a k-protective set. While this is the maximum cardinality k-protective set, the result of that algorithm can also be used to obtain a much smaller k-protective set.

Consider the time t_{end}, when the fire has burned out and every cell is either alive or dead. To find such a time t_{end}, we simply take the maximum $t_i(c) + e_0(c)$ over all cells c with $t_i(c) < \infty$. By definition, all cells are either alive or dead, but none burning at time t_{end}. As the initial set of burning cells S is connected, all dead cells induce a connected subgraph of G. By contrast, the subgraph induced by all alive cells can have several connected components. However, as the initial set of village cells T is also connected, only one of those components contains village cells. We call the set of alive cells from this component, that have at least one dead neighbour in G, the *fire border* B (compare Figure 2.3a).

We call the other vertices from this component the *safe side* of B and all remaining vertices the *fire side* of B. By construction, the cells in B separate the cells on the safe side from the cells on the fire side, i. e., every path from a cell on the fire side to a cell on the safe side contains a cell from B. Consider what changes if we remove the resistance increase of all cells on the fire side of B. In the worst case, all cells on the fire side ignite and burn down. But only the direct neighbours of B affect the survival of cells in B and these ignite and burned even with a resistance increase of k. Thus, removing all vertices on the fire side of B from our largest protective set $V \setminus S$ still results in a protective set for k.

Due to the separation property, for any cell on the safe side of B to ignite, there would first have to ignite a cell on B. Thus, increasing the resistance of vertices on the safe side of k also does not affect whether B survives as a whole. Thus, B is a k-protective set.

Now consider any cell c in B at the end of the fire propagation, i. e. $r_{end} \coloneqq r(c, t_{end})$. Increasing the resistance of c by k did affect its survival if and only if $r(c, t_{end}) \leq k$. Thus, the set of cells c in B with $r(c, t_{end}) \leq k$ is a k-protective set. We will refer to this set as B_{min}.

By construction, B_{min} is the set of minimum cardinality among all k-protective sets that create the same fire border B as $V \setminus S$, but not necessarily among all k-protective sets of G. As illustrated in Figure 2.4, there might be a k-protective set with less cells that creates an entirely different fire border. Using Algorithm IGNITIONTIME we can test if an arbitrary candidate set $P \subseteq V \setminus S$ is a k-protective set. A brute force approach would therefore simply test all $O\left(2^{|V|}\right)$ subsets of $V \setminus S$ and return the minimum one.

There are multiple ways we can do this a little more cleverly. First, if P and P' are both k-protective sets with $P \subseteq P'$, then all cells on the safe side of B_P are also on the safe side of $B_{P'}$. Hence, the subset-relation provides an order on the candidate sets, and $V \setminus S$ creates the fire border closest to the fire with the most cells on its safe side.

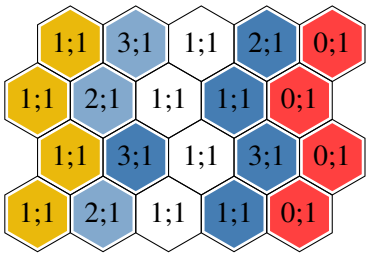


Figure 2.4: A village protection instance with two 1-protective sets creating different fire borders. The initial resistance and energy are given in the cells as pairs of the form $r;e$. The village cells are yellow and the fire cells are red. The fire borders are given by the blue cells, with the respective 1-protective set highlighted. While the right fire border is created by the set of all its cells, the left fire border is created by just a single cell. Thus, this cell forms a Minimum Cardinality 1-protective set.

Another useful observation regards the set B_{min} of minimum cardinality among all k-protective sets that create the same fire border B as $V \setminus S$. Consider an arbitrary cell c and the candidate set $V \setminus (S \cup \{c\})$ with exactly that cell removed. Then candidate set $V \setminus (S \cup \{c\})$ creates a different fire border B' if and only if $c \in B_{min}$. However, B and B' can still share some cells. By construction, if B' is distinct from B, $V \setminus (S \cup \{c\})$ is the set of largest cardinality that creates B'. Let $B'_{min} \subseteq V \setminus S \setminus \{c\}$ be the set of minimum cardinality that creates B'. By the same argument, a candidate set P that creates a fire border distinct from both B and B' must exclude at least one vertex from both B_{min} and B'_{min}. The cells in the minimal set B_{min} are in a sense essential cells for that fire border, and removing any of them guarantees that the resulting set either creates a different fire border or is not a k-protective set at all.

Algorithm 2: TESTSET$((G, r_0, e_0, T), k, P)$

Input : A village protection instance (G, r_0, e_0, T), a protection parameter k and a candidate set $P \subseteq V \setminus S$.

Output : The k-protective set of minimum cardinality among all subsets of P.

```
Run IGNITIONTIME (G, r'_0, e_0) with r'_0(c) = c + k for all cells c ∈ P and r'_0(c) = r_0(c), otherwise.
if t_i(v) = ∞ for all v ∈ T then
    P_min ← the minimal set creating the same fire border as P
    foreach cell c ∈ P_min do
        P'_min ← TESTSET((G, r_0, e_0, T), k, P \ {c})
        if |P'_min| < |P_min| then
            P_min ← P'_min
return P_min
```

This idea is applied recursively in Algorithm TESTSET. As the size of the candidate set decreases with each recursion, it is clear that Algorithm TESTSET terminates. Also, by construction, any set it returns is a k-protective set. To find a minimum cardinality protective set P_{opt}, we need to make at least one recursive call for a k-protective set P that creates the same fire border as P_{min}. When calling

Algorithm TESTSET with $V \setminus S$, the superset of all possible k-protective sets, as the initial candidate set P, it remains to prove, that we do not miss P_{opt}.

Lemma 6
For any k-protective set P, Algorithm TESTSET$((G, r_0, e_0, T), k, V \setminus S)$ *will result in at least one recursive call of Algorithm* TESTSET$((G, r_0, e_0, T), k, P')$ *for a k-protective set P' that creates the same fire border as P.*

By contradiction. Assume the Lemma does not hold for some k-protective sets. Among those, let P^* be the k-protective set that creates the fire border B^* with the most cells on the safe side of it. Let P_{max} be the k-protective set of maximum cardinality, that creates the same fire border as P^*. Let c be an arbitrary cell from $V \setminus S$ not in P_{max}. As supersets of k-protective sets are also k-protective, $P_{max} \cup \{c\}$ is a k-protective set. By the construction of P_{max}, $P_{max} \cup \{c\}$ creates a fire border B_c distinct from B^*. Then, again by the superset argument, B_c has more cells on its safe side than B^*. Finally, as removing c from $P_{max} \cup \{c\}$ creates a different fire border, c must be a part of B^*_{min}.

By the choice of P^*, there must have been a recursive call of Algorithm TESTSET$((G, r_0, e_0, T), k, P')$ for a k-protective set P' that creates the same fire border as $P_{max} \cup \{c\}$. But this would have also resulted in a call of Algorithm TESTSET$((G, r_0, e_0, T), k, P_{max})$, which contradicts our initial assumption. □

It remains to examine the run time of Algorithm TESTSET$((G, r_0, e_0, T), k, V \setminus S)$. At its core, Algorithm TESTSET searches for the k-protective set of minimum cardinality by testing sets of maximum cardinality for every fire border. We can avoid some redundancy by storing the results of previous recursive calls and stopping the recursion, when we encounter a candidate set creating a fire border we already encountered. However, we might still end up with an exponential number of recursive calls, if there are exponentially many distinct fire borders created by k-protective sets. Thus, asymptotically, it might take time $O\left(2^{|V|}|V|\log|V|\right)$ for Algorithm TESTSET to terminate.

Although the asymptotic worst-case might be identical, Algorithm TESTSET$((G, r_0, e_0, T), k, V \setminus S)$ will terminate much faster if there is only a low number of distinct fire borders. This might be due to some structural property imposed on V or the initialization functions r_0 and e_0. If there are only $O(|V|)$ many distinct fire borders B, each will cause at most $O(|V|)$ many recursive calls as $B_{min} \in O(|V|)$ for each B. Then, Algorithm TESTSET$((G, r_0, e_0, T), k, V \setminus S)$ runs in time $O\left(|V|^3 \log|V|\right)$.

2.4.2 The Minimum Fortification Problem

Let (G, r_0, e_0, T) be a village protection instance. Now assume, instead of increasing the resistance of all vertices by k as for the k-protection problem, the firefighters are allowed to increase the resistance of some cells individually before the fire starts to spread. We call a set P a *protective set* if no village cell from T ignites as long as no cell from P ignites. Thus, it suffices for the firefighters to invest in the resistance of the cells in P to protect the cells in T.

As a cell does not ignite if its initial resistance exceeds the sum of the initial energy of all its neighbours, this is equivalent to the following notion: P is a protective set, if and only if $t_i(v) = \infty$ for all village cells $v \in T$ in a modified instance $(G, r'(0), e_0)$, with $r'_0(c) = 1 + \sum_{n \in \mathcal{N}_c} e_0(n)$ for all cells c in P, and $r'_0(c) = r_0(c)$ otherwise.

Now depending on which of its surrounding cells actually ignite, a lower resistance than $1 + \sum_{n \in \mathcal{N}_c} e_0(n)$ might also suffice for a cell P to survive. Instead, the resistance of a cell must only exceed the sum of the energy of exactly those neighbours, that will ignite. Then given any set P we can define its *fortification costs* $c(P)$ as the minimum sum of individual resistance increases for cells in P, such that no cell in P ignites. We will also refer to the individual resistance increase of a particular cell c in P as the fortification costs $c_P(c)$.

Intuitively, T itself is a protective set by definition, and similar to the k-protective sets, every superset of a protective set is also a protective set. However, while sufficient, it is not necessary for a protective set to be a superset of T.

This gives rise to the following optimization problem:

Minimum Fortification
Input: *A village protection instance* (G, r_0, e_0, T).
Goal: *Find a protective set of minimum fortification cost.*

Consider an arbitrary protective set P and the modified instance (G, r'_0, e_0) with $r'_0(c) = r_0(c) + c_P(c)$ for each cell $c \in P$ and $r'_0(c) = r_0(c)$, otherwise. As for the k-protective sets in Subsection 2.4.1, we can consider the state of all cells after the fire has burned out in (G, r'_0, e_0) at some time t_{end}.

Assume, we were given an optimal solution P_{opt} for an instance of the Minimum Fortification Problem, together with the fortification cost for all cells in P_{opt}. By definition, all cells are either alive or dead, but none burning. By the construction of our instance, all village cells and all cells from P are alive due to P being a protective set. As the initial set of burning cells S is connected, all dead cells induce a connected subgraph of G. By contrast, the subgraph induced by all alive cells can have several connected components. However, as the initial set of village cells T is also connected, only one of those components contains village cells. We refer to the set of alive cells from this component that have at least one dead neighbour in G as the *fire border* B. In addition, we refer to the other vertices from this component as the *safe side* of B and all remaining vertices as the *fire side* of B.

As such a fire border separates the village from the fire, it is clear that it is also a protective set. However, for any optimal protective set P_{opt}, the corresponding fire border B_{opt} is also an optimal protective set:

Lemma 7
The fortification cost of P_{opt} is equal to the fortification cost of B_{opt}.

Proof. Adding any cell $c \notin P_{opt}$ alive at t_{end} to P_{opt} does not increase the fortification cost, as they already survive without resistance increase. Thus, the fortification cost of $P_{opt} \cup B_{opt}$ is equal to the fortification cost of P_{opt}.

Let c be an arbitrary cell in $P_{opt} \setminus B_{opt} = (P_{opt} \cup B_{opt}) \setminus B_{opt}$. Obviously, c is alive at t_{end}. If c is in the same connected component as the village, it has no burning neighbours as it is not in B_{opt}, so its fortification cost is 0. If c is not in the same connected component, then its survival does not affect the survival of any village cell. Thus, $P_{opt} \setminus \{c\}$ must also be a protective set. By the optimality of P_{opt}, the fortification cost of c must be 0. As the fortification cost of any cells not in $P_{opt} \setminus B_{opt}$ is 0, the fortification cost of $B_{opt} = (P_{opt} \cup B_{opt}) \cap B_{opt} = (P_{opt} \cup B_{opt}) \setminus (P_{opt} \setminus B_{opt})$ is equal to the fortification cost of P_{opt}. □

While the fortification cost of an arbitrary protective set is not obvious at first glance, the fortification cost of a fire border B is easy to compute. For a cell c in B, let $e_B(c)$ be the sum of initial energies $e_0(n)$ of all neighbours n of c on the fire side of B. As every such neighbour will ignite, the initial resistance of c must exceed the $e_B(c)$. Thus, the fortification cost $c_B(c)$ of a cell c in P is the minimum k, such that $r_0(c) + k \geq e_B(c) + 1$. By definition, the fortification cost of B is the sum of the fortification costs of all cells in B.

So instead of finding an arbitrary protective set of minimum fortification cost, we can search for a fire border of minimum fortification cost. At first glance, this seems to resemble two known graph problems - vertex separator problems and shortest path problems.

For a given graph $G = (V, E)$, a set $S \subset V$ is a vertex separator, if removing S from G splits G into at least two connected components. In our case, the fire border separates the vertices on its safe side from the vertices on its fire side. Research has focused mostly on finding vertex separators of small cardinality (or weight) that split G into connected components of roughly equal size. This was initially motivated to enable efficient "divide-and-conquer" algorithms on graphs [62], but applications, for example in bioinformatics [37] reinforced this motivation more recently. However, we require our separator to separate specific sets of vertices from another (the village and the fire cells). In fact, the set of village cells itself could be the fire border B, in which case B is not even a separator at all as there are no vertices on the safe side. Thus, applying well-known results for vertex separators does not provide a quick solution to the Minimum Fortification Problem.

Shortest path problems are one of the most studied problems on graphs. Given two vertices s and t, a *path* from s to t (or s-t-path) is a sequence of distinct vertices $(s = v_0, v_2, \ldots, v_k = t)$ corresponding to a sequence of distinct edges $(e_1, e_2, \ldots, e_k)$ with $e_i = \{v_{i-1}, v_i\}$. If the sequence of vertices contains duplicates, we call such a sequence an *s-t-walk* instead. A shortest path from s to t is one with a minimum number of edges among all s-t-paths. In the weighted case, a shortest path is instead one with a minimum sum of edge weights. Among the huge number of shortest-path algorithms, often specialized for specific graph or weight types, Dijkstra's shortest path problem is one of the oldest and probably the most famous [28].

Initially, this seems as unpromising as looking at the fire border as a vertex separator, as the optimal fire border B_{opt} does not necessarily need to form a path. Instead, there are instances, where B_{opt} forms a walk, but not a path, an example of which is given in Figure 2.6. However, a fire border

does correspond to a path on the graph formed by the edges and vertices of the regular hexagonal cells inducing G. We call this graph the *border graph* G_b.

Except at the boundary, each vertex of the border graph is adjacent to exactly three cells, and each edge to exactly two cells. Consider the border graph at the boundary of G, as seen for example in the hexagonal graphs in Figure 2.6. We will call the vertices of G_b with exactly two adjacent hexagonal cells the *boundary vertices* of G_b and those with exactly one adjacent cell the *outer vertices*. In contrast, we call the vertices with exactly three adjacent cells the *inner vertices* of G_b. As S and T each form a subpath of the boundary cycle of G, we can split the boundary of G into four sets. Imagine moving counter-clockwise on the boundary cycle starting at some cell in S. We call the set of cells on the boundary cycle between S and T the *upper boundary* and the set of cells between T and S the *lower boundary* of G. In all our examples, where the cells of our smooth hexagonal graph form a rectangle, these are exactly the cells along the lower and upper edge of the rectangle. Equivalently for the border graph G_b, we call the set of boundary vertices of G_b adjacent to cells on the upper boundary of G the *upper boundary* of G_b; equivalently for the lower boundary.

Then all border edges between cells on a fire border B and cells on its fire side form a path π_B from the lower to the upper boundary of G_b. We call such paths *separating paths* in G_b and define their fortification cost $c(\pi_B)$ as the fortification cost $c(B)$ of the corresponding fire border. By construction, a separating path contains no outer vertices at all, and no boundary vertices except the very first and very last vertex. To solve the Minimum Fortification Problem, we need to find the separating path with minimum fortification cost. To reduce this to a shortest path problem in G_b, we want to transform the fortification cost of the path and thus, the corresponding fire border, into edge costs in the border graph G_b, such that the sum of edge costs of a separating path π is exactly the fortification cost.

First, to distinguish which adjacent cell lies on B and which on its fire side, we replace every edge $\{v,w\}$ in the border graph by two directed edges (v,w) and (w,v). Then, for a separating path π from the lower to the upper boundary, the cells on the left side are the cells in the corresponding fire border B, and the cells on the right side lie on the fire side of B.

For an edge e on π we will say it *runs along* the fire border cell on its left side. Now consider the example depicted in Figure 2.5, where $e_0 = 5$ for all cells and cell c has $r_0 = 8$. While it makes sense to charge the fortification cost of c to the edges along c, the fortification cost of c actually changes based on which of the edges along c are included in π. However, as shortest-path algorithms usually build the path incrementally, we will charge the fortification costs of c to the edges in π based on their order in π: The first directed edge of π along c has cost 0 because no additional fortification is necessary to protect c from a single burning neighbour; the second edge has cost 3; every further edge has cost 5, since the resistance of c has to be increased by 5 for every additional burning neighbour.

With such a cost assignment, the cost $c(\pi)$ of a separating path π is exactly the fortification cost of the corresponding fire border B. However, the cost of the same edge e can differ depending on the π it is contained in. But if two paths π_1 and π_2 consist of identical edges up to some edge e_k, the costs of the edges e_1 to e_k is identical in both paths.

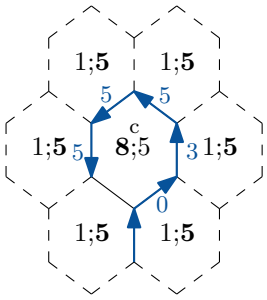


Figure 2.5: The cost of edges on a path in the border graph G_b running along a cell c. The initial resistance and energy values are given in the cells as pairs $r;e$

However, edges along the same cell might not lie on a separating path consecutively, if the corresponding fire border is not a path. As mentioned before, this can even affect the optimal separating path, as illustrated in Figure 2.6. We will say π *revisits* a cell c if π contains a sequence of edges $e_1,\dots,e_k$, where e_1 and e_k run along the same cell c but at least one edge e_i with $1 < i < k$ does not. Hence, the cost of an edge can be influenced by any previous edge in the path, which makes the application of standard shortest path algorithms difficult.

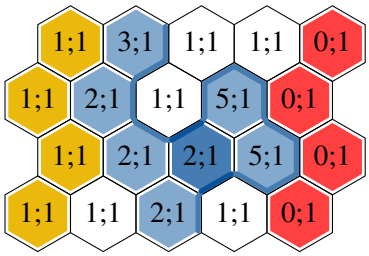


Figure 2.6: A village protection instance, where the fire border (blue cells) of the minimum fortification set is not a path, but a walk. The initial resistance and energy are given in the cells as pairs of the form $r;e$. The village cells are marked in yellow, the cells initially burning are marked in red, while the cells on the fire border are marked in blue. The highlighted cell in the third row is the single cell in the minimum fortification set and has fortificiation cost 1. It is also the cell that is revisited by the corresponding separating path (also marked in blue on the edges of the hexagonal cells).

In the following, we will therefore focus on village protection instances, where the initial energy e_0 is some identical constant f for all cells and the initial resistance of every cells is bounded by $\leq 2f+1$. This implies that, without fortification, each cell can resist at most two burning neighbours, but is always ignited by three. We will subsequently show, that this allows us to disregard paths that revisit cells, when searching for an optimal fire border.

To that end, we will define an alternative cost function that is only locally dependent on the path, but no longer matches the fortification cost exactly. Let e be an edge along a cell c of resistance $r_e := r(c,0)$. We say e is of type r if it comes after a right turn or is the very first edge of path π_b. All edges of this type will run along a different cell than its predecessor on the path. We say e is of type l_k

if it comes after the k^{th} consecutive left turn of π_b along c. Thus, e is the $(k+1)^{th}$ consecutive edge running along the same cell c. A separating path can contain at most five consecutive edges running along the same cell, which limits k to at most four. As we assume all cells to have identical initial energy f, we can define the local cost of an edge e at cell c depending on r_e and its type:

$$lc(e,\text{type}) = \begin{cases} \max(0, f+1-r_e), & \text{if type} = r \\ \min(f, 2f+1-r_e), & \text{if type} = l_1 \\ f, & \text{if type} \in \{l_2, l_3, l_4\}. \end{cases}$$

The local cost $lc(\pi)$ of a path π in G_b is then the sum of the local cost of its edges. Clearly, the local cost of a separating path π in G_b is identical to its fortification cost $c(\pi)$ if it has no revisits, but can be less otherwise. However, given our restrictions on the initial energy and resistance of cells, we can prove the following lemma.

Lemma 8

Let π be a separating path with minimum local cost. Then there exists a separating path π' without cell revisits with fortification cost less or equal to the local cost of π, so $c(\pi') = lc(pi') \leq lc(\pi) \leq c(\pi)$.

Proof. If π is a path without cell revisits, than its local cost is identical to its fortification cost, so π itself fulfils the lemma. So let π be a separating path with minimum local cost with at least 1 cell revisit and let e be the first edge on π that runs along a revisited cell c. Let the edges of c be numbered counter clockwise from 0 to 5, where e is edge 0. Then, the situation at c can be restricted to the following two cases, also illustrated in Figure 2.7.

1 The last edge along c on π is edge 2 and 1 does not lie on π_b.
2 The last edge along c on π is edge 3 and at least one of the edges 1 and 2 does not lie on π_b.

Note that π can neither include edge 4 nor 5. If it included edge 5, π would contain a duplicate vertex and not be a path. If it included edge 4 but not 5, the edge preceding e on π_b would also run along a revisited cell and we assumed e to be the first such edge on π.

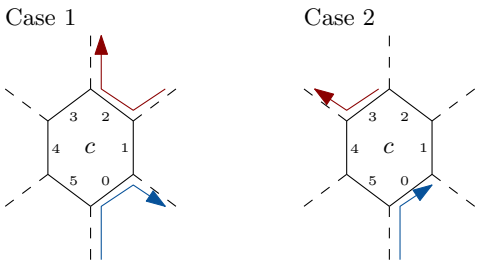


Figure 2.7: Path π follows the blue edges upon its first visit of c and the red edges on its second visit.

We can construct a new separating path π' from π that does not revisit c as follows. In case 1, we can replace all edges on π_b after edge 0 and before edge 2 by edge 1. This increases the local cost of

the path by at most cost $2f$ (f for the new edge 1, and another f, as edge 2 changes its type to l_2). In case 2, we replace all edges on π_b between edge 0 and edge 3 including these two edges by edge 4 and 5. This also adds at most cost $2f$ (f for edge 4 and 5 each). However, the removed part of π can not include more right turns than left as it starts at the lower boundary of G_b and would otherwise intersect with itself. More precisely, as every left or right turn is exactly a 60°-degree turn in the path, we know that in both cases, the removed part contains at least two more l_k edges than r-edges. Thus, the removed part contains at least two l_k-edges with $k > 1$ and thus had local cost at least $2f$.

Hence, the local cost of π' is at most the local cost of π. In addition, π' has strictly fewer edges than π and revisits strictly fewer cells, so after repeating this construction a finite number of times we will arrive at a path π'' without revisits. As π'' has no revisits, its fortification cost is equal to its local cost which is at most as high as the local cost of π which concludes the proof. □

Thus, given our restrictions on the initial energy and resistance of cells, it suffices to find a separating path of minimum local cost. But the local cost is still dynamic in the sense that an edge can have different costs in different paths, albeit no longer dependent on the whole path. This still prevents us from directly applying for example Dijskstra's shortest path algorithm. Due to the locality of the cost definition, if given the information from which neighbour we arrived at a vertex v and which type the last edge had, both the cost and the type of the possible two outgoing edges are fixed. We use this observation to construct a weighted directed graph G'_b from G_b with fixed edge weights, that contains a corresponding path π' for each separating path π without revisits with exactly the local cost of π as the cost of the path π'.

First, we construct the vertex set V'. Consider a triplet (v, p, e), where v denotes an inner vertex v in G_b, the predecessor p a neighbour of v in G_b and e an edge type from the set $\{r, l_1, l_2, l_3\}$. As argued in the proof of Lemma 8, we can exclude edges of type l_4, as they will result in a revisit. Then G'_b will contain 12 copies of the vertex v, one representing each such triplet.

In addition, consider the boundary and outer vertices of G_b. As a separating paths does not contain any outer vertices by definition, we do not need to include them in G'_b. Similarly, we only need to include boundary vertices from the upper and lower boundary of G_b. Of those boundary vertices we need exactly one copy, representing the possible start and end points of our paths. To simplify the application of a shortest path algorithm, we add a universal starting vertex s and a universal end vertex t.

In summary, V' consists of the following vertices.

- A vertex for each triplet (v, p, e), where v is an inner vertex of G_b, p is a neighbour of v in G_b and e is an edge type from the set (r, l_1, l_2, l_3).
- A vertex for each vertex on the lower or upper boundary of G_b.
- A start vertex s and an end vertex v.

Now we construct the edge set E'. For each vertex v on the lower boundary, we add an edge (s, v) with cost 0; and for each vertex v on the upper boundary, we add an edge (v, t) with cost 0.

Each vertex v on the lower boundary has a unique inner vertex neighbour n in G_b, so we add an edge $(v,(n,v,r))$. This edge is of type r and gets the appropriate local cost as edge weight, because it is the first edge of any separating path starting in v.

Now consider a vertex for a triplet (v,p,e), and let the neighbours of the inner vertex v in G_b in counter-clockwise order be p, n_r, and n_l. First, assume n_r and n_l are also both inner vertices of G_b. As the predecessor p signifies from which neighbour a separating path is coming from, any outgoing edge must go to either n_r or n_l. Continuing the path to n_r means taking a right turn, so we add an edge $((v,p,e),(n_r,v,r))$ to our edge set. Similarly, continuing to n_l means taking a left turn, so we add an edge $((v,p,e),(n_l,v,l_k))$ to our edge set, with $k=1$ if the edge type e of the incoming edge is r, and $k=i+1$ if $e=l_i$. If the edge type e in our triple, i. e., the edge type of the incoming edge, is l_3, than taking a further left turn would result in a path with a cell revisit, so we only add the outgoing edge to (n_r,v,r) for those vertices. We know the edge type of all these outgoing edges and can assign the appropriate edge weights. If n_r (or n_l) is a vertex from the lower boundary, we don't add outgoing edges towards that vertex at all, as separating paths will contain vertices from the lower boundary only in the beginning. If n_r (or n_l) is a vertex from the upper boundary, we add an outgoing edge $((v,p,e),n_r)$ of the appropriate edge type as described above.

As a result of this construction, every incoming edge of an inner vertex representing a tuple (v,p,e) will have the associated cost of an edge of type e. As intended, for each separating path of some local cost, there will be an s-t-path of equal cost in G_b'. However, while there exist an s-t-path corresponding to each separating path in G_b, the inverse is not true. If an s-t-path contains two vertices representing two different tuples for the same vertex v, this only corresponds to a walk in G_b, which does not have a corresponding fire border or minimum fortification set. As illustrated in Figure 2.8, this can even affect a shortest s-t-path in G_b'. Thus, finding a shortest s-t-path in G_b' does not necessarily solve the problem of finding an s-t-path of minimum local cost in G_b.

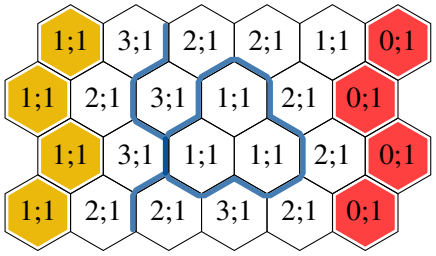


Figure 2.8: A village protection instance, such that the shortest s-t-path in G_b' does visit two copies of the same vertex. The shortest s-t-path in G_b' has cost 0 (and winding number 7), and the corresponding walk in G_b (marked in blue) visits some vertices twice (at the highlighted edge).

However, s-t-paths have an additional property, that we can utilize to avoid this complication.

Let the winding number of a path π in G_b be the difference between the number of edges of type r and the number of edges of type l_i in π. Consider an arbitrary separating path π, starting in a vertex l on the lower boundary and ending in a vertex u on the upper boundary. As all turns in the hexagonal

grid have exactly the same angle, the winding number is a measure of the turn angle of the path. For any right turn, the total angular turn of the path decreases by 60 and every left turn increases the turn angle by 60. Thus, every separating path π^* starting in l and ending in u will have the same winding number as *pi* by the regularity of G_b. We also call this the winding number $w_{l,u}$ of the pair (l,u).

Note that this only applies to pairs of points from the lower and upper boundary. As a separating path starts and ends at the boundary of G_b, it can not have more right or left turns without intersecting itself. Different paths between two arbitrary vertices u and v can have a different winding number.

Now let π be an s-t-path in G'_b containing the vertices l and u from the lower and upper boundary of G_b. Then let π_b be the corresponding l-u-walk in G_b, *i.e.*, the vertex sequence we get by removing s and t from π and replacing every vertex in π representing a triplet (v,p,e) by the corresponding vertex v in G_b. As a reminder, an a-b-walk of length k is a sequence of not necessarily distinct vertices $(a = v_0, v_1, \ldots, v_k = b)$ corresponding to a sequence of not necessarily distinct edges $(\{v_0,v_1\},\ldots,\{v_{k-1},v_k\})$. We call such a walk from the lower to the upper boundary, that corresponds to an s-t-path in G'_b a *separating walk*, similar to the separating paths. We can apply the same notion of winding number to an l-u-walk. Generally, while every separating path from l to u is also a separating walk with a fixed winding number $w_{l,u}$, not every separating walk from l to u with winding number $w_{l,u}$ corresponds to a separating path.

However, given our restrictions on the initial energy and resistance of the cells, we can prove the following lemma.

Lemma 9
Let π_b be the shortest s-t-path in G'_b, such that the corresponding separating l-u-walk in G_b has winding number $w_{l,u}$. This means, for any s-t-path in G'_b strictly shorter than π_b, the corresponding separating l'-u'-walk in G_b does not have winding number $w_{l',u'}$. Then, π_b is a separating path, i. e., π_b does not contain any duplicate vertex.

Proof. Assume, π_b contains at least one duplicate vertex. We will construct a separating path π'_b, such that the corresponding s-t-path in G'_b has lower cost to contradict this assumption and prove the Lemma.

Let $\pi_b = (l = v_0, v_1, v_2, \ldots, v_i, \ldots, v_{k-1}, v_k = u)$. Let $v = v_i$ be the first duplicate vertex, and $v_j = v$ its next appearance. Then consider the path $\pi'_b = (v_0, \ldots v_i, v_j + 1, \ldots v_k)$ obtained by removing all the edges on the walk between v_i and v_j. If the resulting separating walk π'_b still contains duplicate vertices, we can repeat this a finite number of times to obtain a separating path.

However, while removed edges can no longer contribute directly to the local cost of the walk, removing them can change the type and hence the local cost of the edges from v up to the first unaffected r-edge after (v, v_{j+1}). Therefore we must look at the specific situation at v more closely to analyse the change in cost.

The situation at the intersection can take one of the following four cases illustrated in Figure 2.9:

1. π_b turns right at v_i and left at v_j.

2. π_b turns left at v_i and left at v_j.
3. π_b turns right at v_i and right at v_j.
4. π_b turns left at v_i and right at v_j.

Notice that these are all cases because π_b cannot include v_{i-1} a second time as we assumed v_i to be the first duplicate vertex.

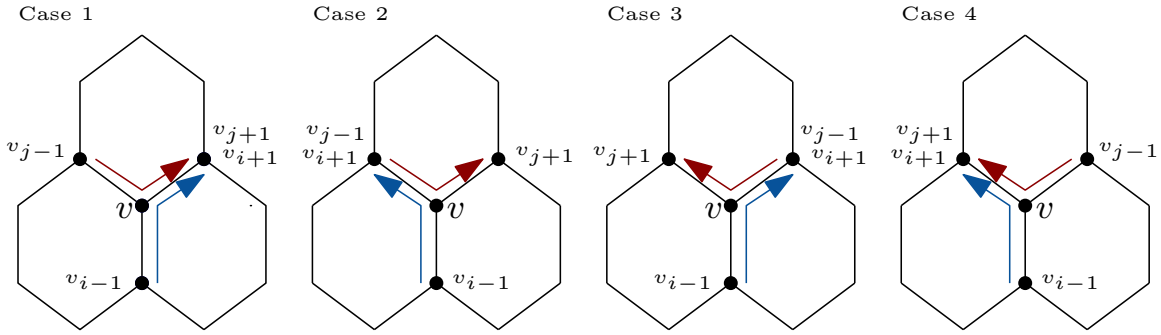


Figure 2.9: π_b follows the blue edges upon its first visit of v and the red edges on its second visit. Cases are equivalent for rotation.

Recall the local-cost function, that dictates the cost of edges in G'_b: edges of type r and l_1 cost between 0 and f; edges of type l_2 and l_3 always cost f. Also for edges along the same cell, edges of type r are at most as costly as l_1 edges and both are at most as costly as edges of type l_2 and l_3. Hence, edges can only become more expensive if they were of type r or l_1 on π_b and the new cost can be at most f.

In cases 1 and 2, (v, v_{j+1}) is an edge of type l_1, l_2, or l_3 on π_b and all further edges until the next r edge are of type l_2 or l_3. On π'_b, (v, v_{j+1}) is of type r, so the local-cost of π'_b can only be less or equal to that of π_b in these cases.

In case 3 and 4, (v, v_{j+1}) is an edge of type r on π_b and (v_{j+1}, v_{j+2}) could be of type l_1, so they could increase in cost by f each. All further edges until the next r edge are of type l_2, l_3, or l_4 and hence do not matter. So π'_b can have local-cost higher than π_b by at most $2f$, if the removed part of the path $v_i, v_{i+1}, \ldots, v_j$ had local cost less than $2f$. But then, the removed part can contain at most 2 edges of type l_k with $k > 1$. But this is not enough to reach the edge (v_{j-1}, v) from (v, v_{i+1}) without taking more right turns than left turns. So in both these cases, the removal of that piece of the path either does not increase the cost of π'_b or the winding number of π'_b is at least 6 higher than that of π_b, because the path took an entire turn with 6 more right turns than left.

But the resulting path π'_b at the end of the repeated removal steps is a separating path from l to u, and therefore a path π'_b with winding number $w_{l,u}$. Therefore, on one of the later removal steps, we must have removed a part of the path, such that the winding number of π'_b decreases by 6. Considering that removing a part changes the type of at most one of the edges still in π'_b to r in any of the cases given above, the removed part contains at least 5 more edges of type l_i than of type r. That means, it must include at least 5 edges of type l_2 or l_3, decreasing the cost of π'_b by at least $5f$, which counters

the cost of the removal that increased the winding number. In fact similar arguments can be made about the removed parts of the path in case 1 and 2, so that in total, the cost of π_b' is strictly less than π_b. □

By Lemma 9, it suffices to find the shortest s-t-path in G_b', such that the corresponding separating l-u-walk in G_b has winding number $w_{l,u}$. To achieve this, we further modify our graph G_b' to another graph $G_b'' = (V'', E'')$ to separate paths by the winding number. Let w_{max} be the maximum absolute winding number of any path in G_b. Then V'' contains $2w_{max}+1$ copies of each vertex v in G_b' (except s and t) representing pairs $(v, -w_{max})$, $(v, -w_{max}+1)$, …, $(v, w_{max}-1)$, (v, w_{max}). E'' contains an edge $((u, w_{max}), (v, w_{max}+1))$, if the vertices u and v in G_b' are connected by an edge (u, v) representing an edge of type r. Equivalently, E'' contains an edge $((u, w_{max}), (v, w_{max}-1))$, if the vertices u and v in G_b' are connected by an edge (u, v) representing an edge of type l_i for some i. The cost of the edges in G_b'' is identical to the cost of the respective edges in G_b'.

By construction, the both $|V''|$ and $|E''|$ lie in $O(|V|\, w_{max})$, where $|V|$ is the number of cells in our underlying smooth hexagonal graph. Trivially $w_{max} \in O(|V|)$, as the winding number of a path can be at most its length. However, a path with very high winding number resembles a big spiral and therefore $w_{max} \in O\left(\sqrt{|V|}\right)$. Thus, Dijkstra's single source shortest path algorithm runs in time $O\left(|V|\sqrt{|V|}\log(|V|\sqrt{|V|})\right) = O\left(|V|^{\frac{3}{2}}\log|V|\right)$ in G_b''.

We still need to actually use G_b'' to find the shortest s-t-path in G_b', such that the corresponding separating l-u-walk in G_b has winding number $w_{l,u}$. If we run a shortest path algorithm in G_b'' starting from some vertex l'' in G_b'' representing a pair $(l, 0)$, where l is a vertex from the lower boundary of G_b. Then we find the shortest path to every vertex u'' representing a pair $(u, w_{l,u})$, where u is a vertex from the upper boundary of G_b. If we do this for every possible starting vertex l'' from the lower boundary, the shortest path among all these corresponds to the shortest s-t-path in G_b' that we seek.

Now consider two separate vertices l_1 and l_2 from the lower boundary. As the winding number is essentially an angle counter, if a separating l_1-u-path has the same winding number as a separating l_2-u-path for an arbitrary u from the upper boundary of G_b, this holds for every such vertex u. We call such vertices l_1 and l_2 winding number equivalent. By the same arguments as for the size of G_b'', there can be at most $O(\sqrt{n})$ many equivalence classes of winding number equivalent vertices from the lower boundary of G_b. For each such equivalence class $[\![l_i]\!]$, we can add a starting vertex s_i to $G''b$ and an edge $(s_i, (l, 0))$ for each vertex l in the equivalence class $[\![l_i]\!]$. Then, we can start our single source shortest path algorithm in each s_i and find the shortest path we seek among the shortest paths to vertices $(u, w_{(l_i,u)})$, where u is a vertex from the upper boundary.

This reduces the number of necessary runs of a single source shortest path algorithm from $O(n)$ to $O(\sqrt{n})$. For some graphs or graph classes, the number of equivalence classes - and hence, necessary runs, might be a constant. For example, in the previous variation of these results as presented at *CALDAM 2020* [58], we assumed the cells to form a rectangle of rows with an equal number of cells each, where the village lies on the left and the initial fire cells on the right side of the rectangle. In this

graph class, which is also used in our examples figures, all vertices on the lower boundary lie in the same winding number equivalence class, such that a single run of a shortest path algorithm suffices.

For general smooth hexagonal graphs, we obtain the following Theorem:

Theorem 4
The Minimum Fortification Problem can be solved in time $O\left(|V|^2 \log |V|\right)$, if each cell has the same initial energy f and the initial resistance of each cell is bounded by $2f+1$.

2.4.3 The Target Ignition Problem

In this section, we will discuss a problem, where instead of trying to protect certain cells as in the previous sections, the goal is to ignite a certain set of target cells T.

Target Ignition
Input: *An instance of the r,e-model (G, r_0, e_0), and two disjoint subsets S and T of V.*
Goal: *Find a minimum cardinality subset $P \subseteq S$, such that igniting every cell in P eventually ignites every cell in T. More precisely, find a subset $P \subseteq S$ of minimum cardinality, such that $t_i(v) < \infty$ for all target cells $v \in T$ in a modified instance (G, r'_0, e_0), with $r'_0(c) = 0$ for all cells c in P, and $r'_0(c) = r_0(c)$, otherwise.*

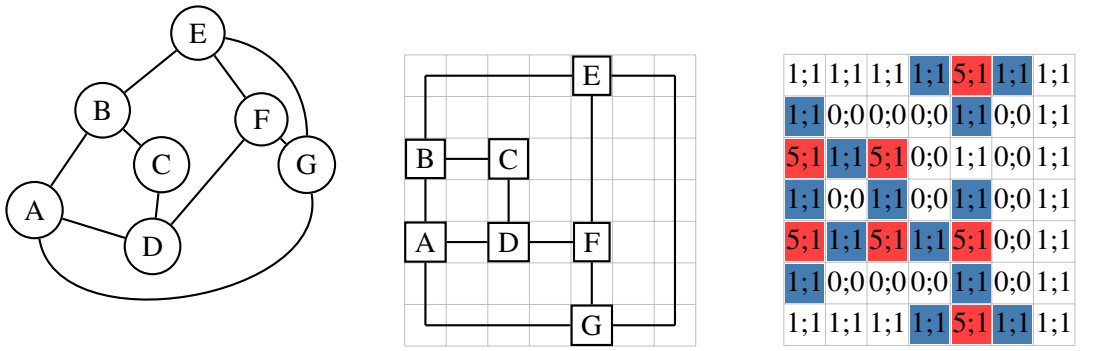
We will prove this problem to be NP-complete, even if G is smooth square graph, by a reduction from planer vertex cover.

Planar Vertex Cover
Input: *A planar graph $G = (V, E)$.*
Goal: *Find a subset C of V that contains at least one endpoint of every edge in E. So for every edge $e = \{u, v\}$ in E, it must hold $C \cap e \neq \emptyset$.*

The planar vertex cover problem has been shown to be *NP*-complete, even for graphs with maximum vertex degree three [39].

To prove the NP-completeness of the target ignition problem, we show how to obtain an instance of the fire expansion problem in polynomial time from a planar graph of maximum vertex degree three. In a first step, we compute a planar grid embedding $G_\square$ of G into the rectilinear grid such that the following holds: Disjoint vertices of G are mapped to disjoint integer coordinates; edges of G are mapped to rectilinear paths in the grid such that no two paths have a point in common, except, possibly, for the endpoints. The size of $G_\square$ is polynomial in $|V|$ and can be computed in polynomial time; see [30].

In a second step, we scale $G_\square$ by a factor of two: a vertex at coordinates (a, b) is mapped to coordinates $(2a, 2b)$; edges of $G_\square$ are stretched accordingly. This introduces *buffer coordinates* between different edges in the embedding of $G_\square$. Finally, we place a square cell at every integer coordinate spanned by $G_\square$ and set the (r, e)-values of the cell c as follows:



(a) A planar graph G (b) Planar grid embedding $G_\square$ (c) Target Ignition Instance for G.

Figure 2.10: An example of the reduction of Planar Vertex Cover to Target Ignition. In the Target Ignition Instance (c), the cells in S are marked red, while the cells in T are marked blue.

- If c corresponds to a vertex of $G_\square$ (vertex-cell) set the weights to $(5,1)$;
- if c corresponds to an edge of $G_\square$ (edge-cell) set the weights to $(1,1)$;
- set the weights of all remaining cells to $(0,0)$.

Add all vertex-cells to S and all edge-cells adjacent to a vertex-cell to T. The size of the resulting instance and the construction time are polynomial in $|V|$ and the resulting graph is a smooth square graph.

It remains to prove that this instance has m cells that ignite all cells in T if and only if G has a vertex cover of size m.

On the one hand, if G has a vertex cover C of size m, the corresponding vertex-cells can be chosen as the set P to ignite. Due to the choice of values, they will ignite all adjacent edge-cells. Since C is a vertex cover, all edge-cells will burn and therefore all cells in T.

On the other hand, assume there is a subset $P \subset S$ of size m that will eventually ignite all cells of T. By construction, all cells of P are vertex-cells, so only vertex-cells are put on fire. Observe that due to the choice of weights, a burning edge-cell can never ignite an adjacent vertex-cell. Moreover, due to the buffer coordinates, any two edge-cells that belong to different edges of $G_\square$ are separated by at least one cell of weight $(0,0)$. Consequently, every edge-cell in T must be either ignited by a directly adjacent vertex-cell or by a fire reaching it from the directly adjacent edge-cell, emanating from a different vertex-cell. Since for every edge in G there is an edge-cell in T, which is ignited via one of the adjacent vertex-cells of P, the vertices of G corresponding to the vertex-cells in P constitute a vertex cover of size m for G.

Certainly the problem is in NP, since the ignition time of all cells can be verified in polynomial time by using the Algorithm IGNITIONTIME presented in Section 2.3. Thus, we obtain the following theorem.

Theorem 5

Target Ignition is NP-complete, even on smooth square graphs.

2.5 Adaptability of the r,e-model

Although energy and resistance of vertices can already model a variety of terrain, we will investigate several different ways to adapt and change the given model, specifically to model further factors affecting forest fires like wind and vegetation height. Most of the time, these adaptations are achieved by slight changes or additions to the transition rules and can often be combined to model several different factors at once. We will also discuss how these changes affect the complexity of this model, especially but not limited to its effect on the runtime of the snapshot algorithms discussed in Section 2.3.

2.5.1 Heat and Combustion Zones

In the r,e-model the energy parameter only governs how long a vertex burns once ignited, but different areas of the forest might also burn with varying temperatures, igniting adjacent areas more or less easily. An intuitive way to adapt the given r,e-model accordingly, is to give vertices additional weights to model how fast and hot a vertex burns with more precision.

As an intuitive way to implement this, we can give each vertex v an additional constant weight parameter h representing the heat of the vertex when burning. To reflect, that hotter vertices ignite their neighbours more easily, the transition rule for the resistance of alive vertices changes to:

- If v is *alive* at time t then $r(v,t+1) \coloneqq \max\{r(v,t) - \sum_{u \in B} h(u), 0\}$, where B is the set of burning neighbours of v at time t.

This change adds only a constant factor to the computation of a single vertices transition in a round and the sweep-subroutine used in the Algorithm IGNITIONTIME and hence does not affect their asymptotic runtimes.

Taking this idea a small step further allows us to address a mayor drawback of many firefighting models, that Pastor et al. bring forth in their survey [71]. They observe that the spread fire is often modelled as a single line moving forward instead of taking combustion zones into account. The desired effect would be that a fire front with a lot of forest still burning behind it spreads faster than on its own. Instead of using a constant heat parameter $h(u)$ in the rule above, the heat of a vertex could be the number of its burning neighbours, reflecting that a vertex with a lot of fire around it burns hotter itself.

This does not affect the asymptotic performance of the step-by-step simulation beyond a constant factor. It suffices to go over all edges each round to count the burning neighbours of each vertex to determine its heat. But a single round of simulation takes time $O(|E|)$ anyway to compute the adjusted resistance values. However, as a vertex starting (or ending) to burn does not only affect its direct neighbours, but also the neighbours of these neighbours, the preprocessing based approach is affected more. Both the number of times a vertices ignition time is recomputed and the number of vertices affecting said recomputations increases. However, even if all vertices are affected in each round of

the algorithm, and each vertex affects each other vertices ignition time, the resulting algorithm still runs in $O\left(|V|^3\right)$. When restricted to graphs with the maximum vertex degree bounded by a constant, the number of vertices affected by a single vertex is constant, so the asymptotic runtimes remain unchanged.

2.5.2 Wind

Wind is one of the biggest factors affecting the spread of wildfires. Depending on the strength of the wind, a fire might spread at different speed or only in a specific directions. Wind and weather conditions might also change while the fire is spreading.

One general way to introduce wind to the r,e-model is to replace the undirected input graph $G=(V,E)$ with a directed graph $G'=(V,E')$ with two directed edges $(u,v),(v,u)\in E'$ for each edge $\{u,v\}\in E$. Then we can assign each edge (u,v) a positive integer weight $w(u,v)$ representing the strength of the wind along this edge, i. e., how strongly it affects the resistance of v if u is burning. The transition rule for the resistance of alive vertices intuitively changes to:

- If v is *alive* at time t then $r(v,t+1) := \max\{r(v,t) - \sum_{u\in B} w(u,v), 0\}$, where B is the set of burning neighbours of v at time t.

These changes add only a constant factor to the computation of a single vertices transition in a round and the sweep-subroutine used in Algorithm IGNITIONTIME and hence does not affect their asymptotic runtimes. Even allowing the wind, and thus, the weight w to change a constant number of times during the fires propagation adds only constant overhead.

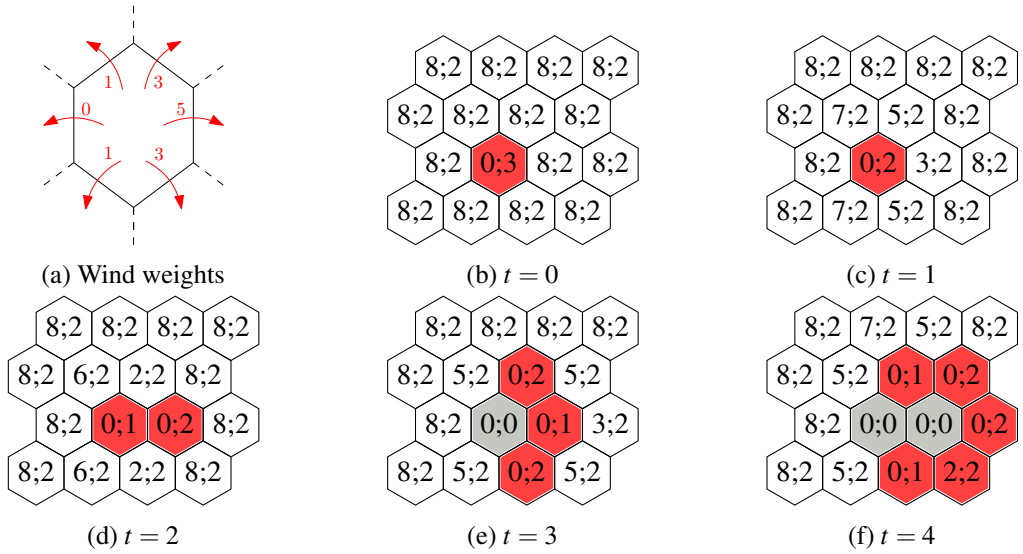
When restricting to a graph induced by a regular tessellation like the regular hexagonal or square tiling, it can make sense to have a global wind setting affecting all cells identically. Figure 2.11 gives an example of how a strong west wind would be represented for a hexagonal cell graph.

2.5.3 Ground and Crown Fires

A forest's greenery and ecosystem can be separated into multiple levels from the roots up to the canopy and each level might provide different conditions affecting the spread of a fire. For example, depending on the type of forest, there might be different amounts of low bushes and dry timber near the ground, and the thickness of the canopy can vary. Thus it makes sense to consider the spread of the fire separately for the different levels as well as how it spreads from level to level. Graham et al. [42] classify fires into different types depending on what level of the forest it is burning through.

Ground fires burn closest to the ground, mainly feeding on moss, lichen and duff. Due to the wetness and thickness of the fuel, they are often spreading very slow and mostly smoulder rather than burn with open flames. Often a remnant of higher forest fires, they can keep burning over weeks and even months and sometimes grow to surface fires.

Surface fires are fires feeding on low bushes and dry timber near the ground. These fires usually spread more freely in forest consisting mostly of broad-leafed trees, as the lighter canopies facilitate



(a) Wind weights (b) $t = 0$ (c) $t = 1$

(d) $t = 2$ (e) $t = 3$ (f) $t = 4$

Figure 2.11: The first figure denotes the weights representing a strong west wind in each direction. The following figures show an example how a fire spreads through an instance according to this wind. While all cells have the same resistance, the fire spread much faster in the eastern direction.

the growth of more surface fuel like bushes. Alternatively, a tree that has recently died and is especially dry, or a young tree with some low-hanging branches can act as torch for a surface fire to ignite a crown fire.

Crown fires are fires burning mainly through the canopy of a forest. These usually spread fast and are especially susceptible to the effect of wind and weather, which fires on the lower level are partially sheltered from.

When using a graph induced by a regular tessellation to model the terrain, one can intuitively stack multiple copies of the graph on top of each other to represent these layers. The different varieties in spreading speed and burn time could then be nicely simulated with the resistance and energy parameters. The resulting graph would still have the degree of each vertex bounded by a constant and no further effect on the runtime of Algorithm IGNITIONTIME.

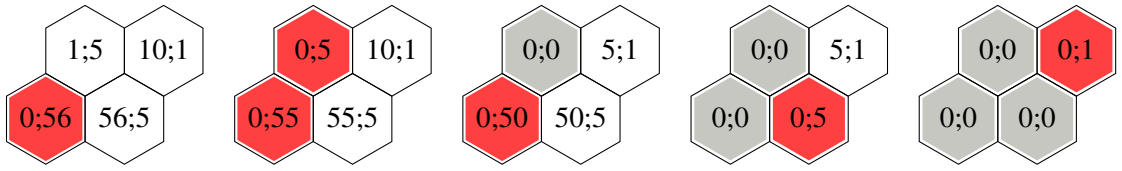
If canopy and surface are especially far apart, it can make sense to consider an additional ladder level between surface and crown representing fuel like vines growing up tree trunks or a half-toppled tree leaning against another. In contrast to the other levels however, such ladder fuel mostly facilitates surface fire spreading to crown fires without much spread within the layer itself. In that case in particular, one might want to consider different rules for spreading between levels and within. This can be implement using weighted (directed) edges as proposed for modelling wind in Subsection 2.5.2.

But even without any changes to the transformation rules of the model, stacking a small constant number of layers admits a surprising power of computation if one considers infinite graphs. As David Kübel proves in his PhD-Thesis[46], two layers of the infinite square grid graph with constant description complexity with respect to r_0 and e_0 allow to simulate any Turing machine. The core idea

is to show that the problem of whether a cell c in an instance (G, r_0, e_0) ignites can be reduced to the halting problem.

2.5.4 Regeneration and Regrowth

The resistance of a vertex being reduced by burning neighbours can be imagined as it slowly heating up until it is finally hot enough to ignite. This visualisation highlights one drawback of our model. While it is nice that the effect of multiple burning neighbours is accumulated towards igniting a vertex, these neighbours need not be burning at the same time, as the effect of a burning vertex on the resistance of its neighbours is permanent. This can lead to scenarios as illustrated in Figure 2.12, where a vertex is ignited by the joint effect of two neighbours, that each burn for just five rounds, but do so 50 rounds apart.



(a) $t = 0$ (b) $t = 1$ (c) $t = 6$ (d) $t = 56$ (e) $t = 61$

Figure 2.12: An example of a cell that is ignited by two neighbours burning 50 rounds apart. The initial resistance and energy are given in the cells as pairs of the form r;e.

To counter this effect, one can allow alive vertices to regain resistance up to either the initial resistance or a separately defined upper limit $r_{max}(v)$ as long as none of its neighbours is burning. Allowing the same for both the resistance and the energy of dead vertices simulates the regrowth of greenery after a fire. Whether it is realistic to model this within the same time frame as the spread of the fire is questionable, but this might be a better fit better when using this model for something different than fire. In this case r_{max} must be greater than 0 for all v, otherwise a dead vertex could regenerate a point of energy but not resistance and immediately ignite again.

These changes affect propagation rules for alive and dead vertices as follows:

- If v is *alive* at time t, then $e(v,t+1) := e(v,t)$. If v has burning neighbours, then $r(v,t+1) := \max\{r(v,t) - b, 0\}$, where b denotes the number of burning neighbours of v. Otherwise, $r(v,t+1) := \max\{r(v,t)+1, r_{max}(v)\}$.
- If v is *dead* at time t and has burning neighbours, then $r(v,t+1) := r(v,t) = 0$ and $e(v,t+1) := e(v,t) = 0$. Otherwise, $r(v,t+1) := \max\{1, r_{max}(v)\}$ and $e(v,t+1) := \max\{1, e_{max}(v)\}$.

As these changes to the transformation rules allow vertices to change their state infinitely often, it is clear that this significantly increases the computational power of our model. When restricted to a regular square grid, it can be considered as a cellular automaton of class 4 (compare Subsection 2.2.2). By the same reduction as for the layered version, David Kübel proves in his PhD-Thesis, that an infinite square grid graph with constant descriptions complexity with respect to r_0 and e_0 allows to

simulate any Turing machine using these additional rules. This fits Wolfram's conjecture, that cellular automata of class 4 are capable of universal computation.

On the other hand, even in finite graphs, a vertex might ignite, burn down, regrow and ignite again over the course of several rounds. This means, that the snapshot algorithm introduced in Section 2.3 can not be applied to this variation, as it is based on determining the unique ignition time of very vertex as a preprocessing step before computing the snapshot at a given time t. Although it can be adapted to cope with this in general, its termination and runtime then must depend on t for its termination, losing its advantage in comparison to the naive step-by-step simulation.

The only way to remedy this might to the detect a regular recurrence of snapshots. Due to the upper limit on the resistance and energy, every vertex can only attain a finite number of states. That means, there is only a finite number of state combinations for all vertices in a finite graph, i. e. a finite number of possible snapshots. As the transition rules are deterministic, if the snapshots at times t and t' are identical with $t' > t$, it holds:

- The snapshot at time $t^* = t' + (t' - t)$ is identical to the snapshots at times t and t'.
- The snapshots at time $t + k$ and $t' + k$ are identical for any $k \in \mathbb{N}$.

2.5.5 Continuous Time

Another adaptation of the r,e-model that suggests itself is to move from a discrete to a continuous time model. While moving the whole model of fire propagation from a discrete to a continuous domain creates a new family of problems more closely discussed in Chapter 3, considering just the flow of time as continuous instead of discrete time steps changes surprisingly little about the model.

The definitions of *alive*, *burning* and *dead* stay the same. However, the step-wise transformation rules for the resistance and energy are replaced by non-increasing, continuous, piecewise-linear functions. In an interval, during which the number of burning neighbours b of an alive vertex v is a constant, it is a linear function with slope $-b$. From when the resistance of v arrives at 0 at some ignition time $t_i(v)$, the resistance function remains constant. Note that the coefficients of all piece-wise linear functions are integers, so all ignition times will be rational.

The energy function $e(v,t)$ on the other hand is constant until $t_i(v)$, followed by one piece of a linear function with slope -1 until it arrives at 0 as well. From then on, both the energy and resistance function remain constant.

Given the same instance (G, r_0, e_0), this does not affect whether a vertex will ignite or not, as a single neighbour will effectively reduce a vertices resistance by at most its initial energy value – the same as in the original model. However, the ignition time in this continuous model will always be at most as big as the ignition time in the discrete time model, as we essentially just remove rounding to the next integer. While the step-by-step simulation algorithm for snapshot computation is no longer applicable to this continuous time model, the preprocessing based approach works just fine with very slight adjustments.

2.6 Open Questions

As discussed in Section 2.2, the original Firefighter problem is just as hard in the r, e-model as in Hartnell's original model. Thus, it suggests itself to investigate other problems and questions one can pose on the foundation of this new model. Besides those discussed in Section 2.4, we will present some additional problems to study, focusing on some that utilize more unique features of the r, e-model, e. g. the fact, that the fire does not necessarily spread to every vertex of the graph even without any firefighting measures.

2.6.1 Burning Number

Consider the burning number of a graph as introduced by Bonato et al. [12]. Given a graph G, one chooses one vertex to be ignited each round in addition to the neighbours of already burning vertices. The minimum number of rounds it takes with any sequence of vertices to burn the whole graph is the burning number $b(G)$.

This notion can be adapted for an instance (G, r_0, e_0) with $r_0(v) > 0$ for all vertices, such that no vertex is initially burning. Then, igniting a vertex is equivalent with reducing it's resistance to 0. As igniting a single vertex might not be enough to ignite the whole graph, one might then ask several questions:

- What is the minimum number of vertices, such that igniting them ignites the whole graph?
- If one is allowed to reduce the resistance of a single vertex to 0 each round, what is the minimum number of rounds it takes to burn down the whole graph?
- If one is allowed to reduce the resistance of a single vertex by 1 each round, what is the minimum number of rounds it takes to burn down the whole graph?

The answer to these questions is dependent on both the structure of the graph and the resistance and energy values of the vertices. Answering the first question is actually a special case of the problem discussed in Subsection 2.4.3 where both the set of source and the set of target cells is the whole vertex set.

2.6.2 Evacuation Distance

The aspect that some parts of a graph might remain unburned allows us to propose problems regarding the evacuation to safe areas. Imagine a person starts at an arbitrary vertex v in round 0 of an instance (G, r_0, e_0) and can move along an edge to an adjacent vertex each round. An evacuation path would be a path from v to a safe vertex of G that will not ignite, i. e. has ignition time ∞, such that a person moving along that path never ends a round in a burning vertex. We say a vertex v has evacuation distance k, if there exists an evacuation path of length k starting in v, and evacuation distance ∞, otherwise.

We can identify all vertices with finite evacuation distance by starting with the safe vertices. Using the fire propagation preprocessing algorithm discussed in Section 2.3 to compute the ignition time of each vertex, we can identify all vertices with ignition time ∞. These are all the vertices with an evacuation distance 0. A person starting on a neighbour of such a vertex can evacuate within a single round, except when it's vertex was already burning at round 0. Thus, a vertex that did not ignite at round 0, and has a neighbours with evacuation distance 0, has evacuation distance (at most) 1. By the same argument, vertices that are still alive in round k, and have a neighbour with evacuation distance k, have evacuation distance (at most) $k+1$. By these arguments, a single Breadth-First-Search from the safe vertices suffices to determine the evacuation distance of all vertices in G.

A firefighter might be especially interested in those vertices that do not have an evacuation path. With saving as many people as possible as a priority, a firefighter would want to maximize the number of vertices that have evacuation paths. To do that he could protect some vertices to create new vertices to evacuate to, which gives rise to the following problem:

k-Evacuation Spot Problem
Input: *An instance of the r,e-model.*
Goal: *Find k vertices, such that protecting these vertices maximizes the number of vertices with evacuation distance $< \infty$.*

2.6.3 Budgeted Firefighting

A direct extension of Hartnell's Firefighter as described in Section 2.2 is at least as hard as the original problem. However, the resistance and energy values of the vertices allows to define firefighting actions in a more nuanced way than just protecting a vertex entirely. Generally, two types of actions reduce the spread of the fire: On the one hand, increasing the resistance of vertices not yet on fire can cause them to ignite later or not at all, On the other hand, decreasing the energy of vertices, which already did or eventually will ignite, reduces their impact on adjacent vertices, causing them to ignite later or not at all.

So instead of protecting a vertex each round, the firefighter can increase the resistance or reduce the energy of one or more vertices, with the sum of those changes limited by some budget b. The goal can also vary between minimizing the number of burned vertices, the number of rounds until the fire burns out, or even the number of vertices affected by the firefighters or the fire. The last option would be motivated by the fact that firefighting efforts themselves can cause damages, so we might want to apply them efficiently.

When reducing the objective to protecting a set of target vertices, we arrive at more general dynamic versions of the village protection problems discussed in Section 2.4, where increasing the resistance of the protective sets takes time. This will result in protective sets closer to the village cells to be better, as the distance to the fire allows for more time to apply the resistance increases. In addition, it might be necessary to fortify some cell closer to the fire, that is not part of the final fire border, to delay the fire, such that the firefighters have enough time to fortify the actual protective set.

Note however, that the results we obtained for the static versions in addition heavily utilize topological properties of the fire border in the underlying graph and set of village cells and require restrictions on the possible initial energy and resistance values.

A plethora of problem variations also arises by varying the exact parameters of protection. The firefighters might be allowed to affect only resistance, only energy or both. In the last case, there might be separate budgets for both types of actions; or a joined one, but affected at different weights by the two types of action. For example, preventative measures (i. e., increasing resistance) might be cheaper than actually extinguishing a fire (i. e., decreasing the energy of burning cells).

Besides adapting the exact parameters, more specific variations can also be obtained by focusing on certain graph classes, or simplifying conditions on the initial resistance and energy functions. Based on the complexity of Hartnell's Firefighter Problem, as well as the challenges encountered in the static version of the village protection problems, the complexity of these types of dynamic firefighting problems might depend heavily on exploiting some regularity in the input instance.

Chapter 3

Continuous Firefighting

In this chapter we study the effect of delaying barriers in continuous firefighting.

We begin by introducing the canonical problem variant as introduced by Bressan in more detail. We shortly present both the best known upper and lower bound on the necessary building speed v and argue how the lower bound proof motivates the investigation of delaying barriers. We then present three smaller related problems, in which delaying barriers play three distinct roles.

Afterwards we present the problem variant of continuous firefighting considered in detail in this chapter - to contain the fire in a half-plane. We present some structural observations on optimal strategies that admit some basic lower bounds on the building speed v and subsequently refine these using three different approaches resulting in three different lower bounds, the best of which shows $v \geq 1.66$ is necessary. We present an approach to construct a working strategy for $v < 2$ and subsequently optimize this approach to create a strategy for which a speed of $v > 1.8772$ suffices.

Finally, we discuss some possible extension of our results including some open questions.

3.1 Bressan's Original Problem

Originally, Bressan proposed a rather general model for geometric firefighting, in which the starting area is some arbitrary subset R_0 of the plane and the spread of the fire is governed by a multifunction $F : \mathbb{R}^2 \longmapsto \mathbb{R}^2$ meaning the fire spreads with different speed in different directions. The area $R(t)$ reached by the fire at some point in time t is then modelled as the reachable set of a differential inclusion, a concept defined in [4]. For a point $p \in R(t)$ there must exists a path from a point in R_0 to p, such that the time it takes the fire to move along that path according to the spread function F is at most t.

Similarly, strategies are defined by set-valued maps $t \mapsto \gamma(t) \subset \mathbb{R}^2$, where $\gamma(t)$ is the set of all barriers built until time t. For such a function $\gamma(t)$ to describe a valid strategy for a building speed v, it must fulfil two conditions: Firstly, as barriers can not be demolished or moved, it must hold $\gamma(t_1) \subseteq \gamma(t_2)$. Secondly, to not exceed the building speed limit, the set $\gamma(t)$ must consist of curves of measurable length and that length may not exceed $v \cdot t$ at any time t.

The area reached by the fire not crossing the barriers built according to γ is called $R^\gamma(t)$. Again, for a point $p \in R^\gamma(t)$ there must exist a path from a point in R_0 to p that the fire can follow in time t without conflicting with γ. That means, if it takes the fire time t_x to reach a point x on that path according to F, then $x \notin \gamma(t_x)$. The fire is contained by a strategy $\gamma(t)$ if $R^\gamma(t)$ is ultimately bounded, i. e. if $R^\gamma(t) \subseteq B_r$ for all $t > 0$ and some fixed ball B_r with radius r centered at the origin.

Bressan also made one central observation that simplifies the analysis of strategies significantly: Instead of defining a strategy dynamically as a function $\gamma(t)$ fulfilling the conditions given above for some building speed v, one can also equivalently define a strategy by its static end result $S \subset \mathbb{R}^2 = \lim_{t\to\infty} \gamma(t)$. We call such a set a *barrier system*. The reachability set of the fire $R^S(t)$ then contains all points for which S-free reachability paths of length at most t exist.

Let $S \subset R^2$ be a set of barriers such that $R^S(t)$ is bounded, i. e. S contains the fire. Then the barriers in S only affect the spread of the fire from the moment they are reached by it. It is therefore intuitive to build the barriers as fast as possible in the order they are reached. If the total length of barriers in S reached by the fire at any time t does not exceed $v \cdot t$ for a building speed v, this approach yields a dynamic strategy γ with $R^\gamma(t) = R^S(t)$. The speed necessary to build every barrier in S at the latest right before the fire reaches it, is exactly the lowest possible speed v for which $R^\gamma(t) = R^S(t)$.

Bressan's results nearly exclusively focus on the special case where both R_0 and $F(x)$ are the unit circle. This means that the fire spreads at unit speed in all directions starting with the unit circle at $t = 0$. We call this the *canonical geometric firefighting model*. In that variant, the time it takes the fire to reach any point is just the length of the euclidean shortest path from R_0 to that point.

Regarding the necessary speed v, they obtained two main results for the canonical variant.

Theorem 6: Bressan's Upper Bound[14]
There exists a strategy containing the fire in the canonical geometric firefighting model *for any speed* $v > 2$.

The strategy they consider is the union of two logarithmic spirals. Starting at the boundary of the fire and the building with constant speed v just along the edge of the fire produces a logarithmic spiral with excentricity $\alpha := \cos^{-1} 1/v$ for any speed $v > 1$. Hence, with speed $v > 2$ one can start two such spirals in opposite directions from the same point on the unit circle at speed $v/2$ each. After half a round both spirals will meet opposite their starting point and hence the fire is contained.

Theorem 7: Bressan's Lower Bound[14]
There exists no strategy S containing the fire in the canonical geometric firefighting model *for any speed* $v < 1$.

Although Bressan first proved this in [14], we will give the much simpler proof from his survey [16] based on the ideas first used in a slightly different setting (compare [17]).

Proof. Assume, there existed a strategy given as a set $S \subset R^2$ that contains the fire for some speed $v < 1$. Then, S must contain some outer barrier S_O separating the starting area R_0 from the rest of the

plane, and may contain some inner delaying barriers S_I attached to S_O and some free-floating barriers S_F inside the area bounded by S_O (compare Figure 3.1). Any barriers outside of S_O do not affect the spread of the fire and can therefore be disregarded. Let $|S_I|$, $|S_O|$, $|S_F|$ be the sum of the lengths of all barriers in these respective sets.

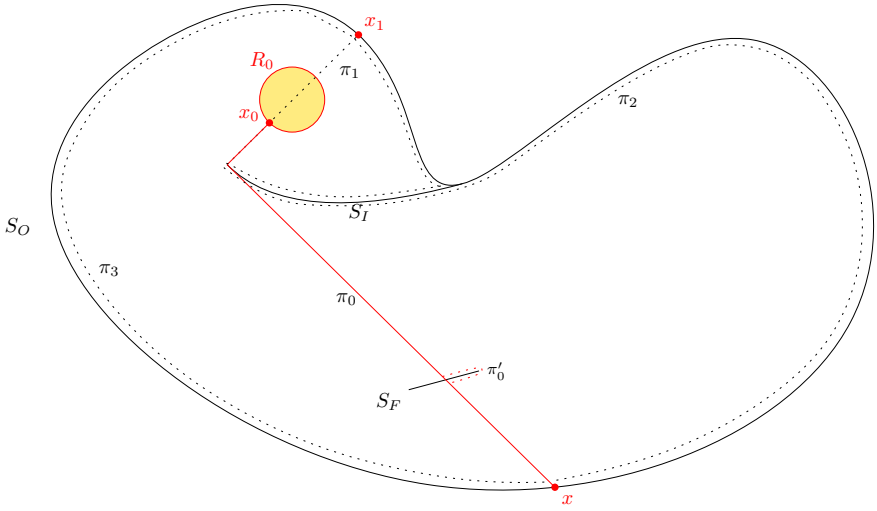


Figure 3.1: An illustration of the lower bound proof construction.

Let x be the last point on S_O reached by the fire, the 'last brick' in the wall. We will ignore the free-floating barriers for now and handle them in the end. Consider the euclidean shortest path π_0 from R_0 to x not intersecting any barrier from S_I. It starts in some point x_0 on the boundary of R_0. Consider prolonging this path from x_0 in a straight line until it hits S_O at some point x_1.

Now we can compare three paths from x_1 to x. Let π_1 be the path via x_0, and π_2 and π_3 be the paths clockwise and counter-clockwise along S_O and S_I. By construction $|\pi_0| \leq |\pi_1| \leq \min(|\pi_2|, |\pi_3|) \leq 1/2\,(|\pi_2| + |\pi_3|) = 1/2\,(2S_I + S_O) \leq |S_I| + |S_O|$. Now assume π_0 crosses a free-floating barrier $b \in S_F$. Then we can construct a path from x_0 to x not crossing b by following π_0 and when hitting b going either left or right along b until we hit π_0 again, depending on what's shorter. This adds a detour of at most b to the path. Let π_0' be a path from x_0 to x not crossing any barrier from S constructed in that manner. Then $|\pi_0'| \leq |\pi_0| + |S_F| \leq |S_I| + |S_O| + |S_F|$.

This means, that the fire reaches x_0 before time $t = |S_I| + |S_O| + |S_F|$, i. e. before we could have built all barriers in S with building speed 1, which contradicts our original assumption and concludes the proof. □

Note that, if we could prove that there exists an optimal strategy, that does not contain any inner delaying or free-floating barriers, the same arguments immediately hold for $v < 2$. Hence a closer look at such inner or free-floating barriers is justified.

3.2 Related Problems Showcasing Delaying Barriers

We will now present three smaller related models, for which the best known strategies utilize delaying barriers in three distinct ways.

The single barrier model is very similar to the canonical geometric firefighting model with a slightly weaker firefighter. Instead of being allowed to build barrier curves freely in the plane, a single continuous barrier curve has to be built. The best known strategies in this variant build the barrier in spiral coils around the fire area until closing onto itself after several rounds for a building speed $v > 2.6144$. Hence, a big part of the barrier curve, namely all spiral coils except the last one form a single long delaying barrier while the last coil forms the enclosing outer barrier.

The angle cover model is similar to the canonical geometric firefighting model, but has a weaker fire. Instead of spreading at unit speed in all directions from the burning area of the plane, the fire spreads in straight lines from an origin point, burning everything visible from the origin similar to a nuclear flash. This means that the fire does not burn around corners or endpoints of barrier curves, such that strategies to contain the fire just need to block every angle, i. e. intersect every outgoing ray from the origin. As a barrier either blocks a ray or has no effect at all, an optimal strategy for this model contains no delaying barriers at all and blocks the fire for any speed $v > 0$.

The barrier following fire model is closely related to the main model studied in this chapter, where the fire is to be contained in the upper half-plane, but has a weaker fire. The fire starts in the middle of a small piece of barrier at the origin and is restricted to only burn along the barriers instead of freely through the plane. The firefighter is restricted to barrier systems consisting of the horizontal x-axis and vertical barriers attached to it and has to prevent the fire from ever reaching any part of the horizontal barrier from below. Hence the whole horizontal barrier is the outer enclosing barrier that limits the fire to the upper half-plane while the vertical barriers are delaying barriers. As the fire burns twice along each vertical barrier - first up and then down again - building as many of them as possible is crucial to effective strategies for this model. To avoid degenerate strategies that just build vertical barriers at infinitesimally short distances to each other, we require that a vertical barrier of height b has distance b to the next vertical barrier in the same direction. The best known strategy fulfilling this requires a building speed of $v > 4/3$ to contain the fire.

3.2.1 The Single Barrier Model

An intuitive variant of the geometric firefighting problem is to allow just one continuous barrier curve that has to built in one continuous move. This can be interpreted of just a single firefighter moving through the plane deploying the barrier in its path. We call this the single barrier model. Such strategies are a subset of the strategies for the canonical variant, hence the lower bound of $v \geq 1$ (compare Section 3.1) still holds. Focused again on the canonical fire with unit spread in all directions and the unit circle as its starting area R_0, this variant has been discussed by both Bressan [21] and more in depth by Klein et al. [57].

Both show the following theorem by providing slightly different strategies.

Theorem 8: Single Barrier Upper Bound
The fire can be contained in the single barrier model with building speed $v > 2.6144...$.

Both strategies form a barrier curve starting at the fire boundary, spiralling around the fire multiple rounds based on different methods before eventually closing in on itself, such that the last coil of the spiralling curve hits the previous one to contain the fire. This means, that the only the last coil of either barrier curve actually forms the enclosing outer barrier and all previous coils form one long delaying barrier slowing the fire just enough that the barrier curve can be closed.

However, this delaying part of the barrier curve is absolutely crucial to the success of the strategies. When trying to design a strategy without delaying barriers, the most simple one would be just building a circle at some distance to the origin, which requires speed $v > 2\pi$. During a presentation at SoCG 2015 on a preliminary version of the paper by Klein et al. [53], Langetepe showed a more clever construction using just a single round which is illustrated in Figure 3.2. It starts at a point q_0 away from the fire and builds a line segment in such a way, that it coincides with the growing fire boundary at a point q_1. From q_1 one the barrier curve follows the logarithmic spiral centered at the fire origin, that runs just along the boundary of the fire, until it intersects the line segment exactly back at the starting point q_0. This strategy without delaying barriers works for $v > 3.7788$ which is still much worse than either spiralling strategy.

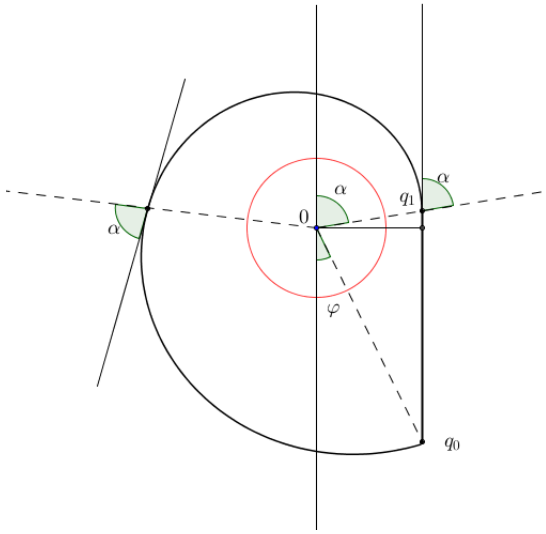


Figure 3.2: A strategy to contain the fire within a single round.

Bressan's spiralling construction is based on the following observation. Let P be a point on the logarithmic spiral $S^\alpha = (\varphi, e^{\varphi \cot \alpha})$ of excentricity α, and let Q denote the next point on S^α touched by a tangent at P; see Figure 3.3, (i).

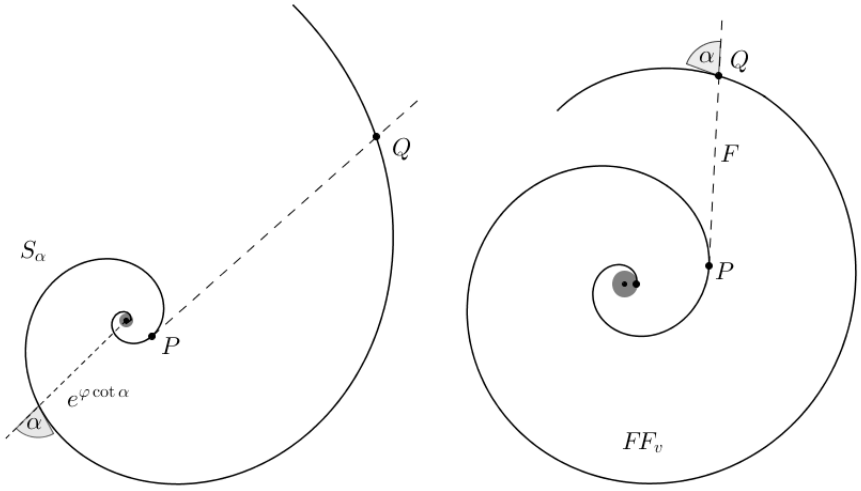


Figure 3.3: (i) A logarithmic spiral S_α of excentricity α; angle φ ranges from $-\infty$ to ∞. (ii) Curve FF_v results when the fighter moves at speed v along the fire's expanding boundary.

Then the spiral's length to Q is at most $2.6144\ldots$ times the sum of its length to P plus the length $|PQ|$ of the tangent, for all values of α. In other words, if the fighter builds such a barrier at some speed $v > v_c := 2.6144\ldots$ he will always reach Q before the fire does, which crawls around the spiral's outside to point P and then runs straight to Q. In [21] the fighter uses this leeway to build a sequence of logarithmic spiral segments of increasing excentricities α_i that stay away from the fire, plus one final line segment that closes this barrier curve onto itself.

The strategy studied by Klein et al. [57] instead is to construct a barrier curve FF_v by starting at some point p_0 on the boundary of the fire starting area and continue to build just along the spreading fire. This means, that at any point p on the curve, the length of the barrier curve up to that point is always exactly v times the length of the shortest fire path to that point, i. e. the shortest path from the fire origin area to the point p. In the beginning this forms a logarithmic spiral around the origin with excentricity $\alpha := \cos^{-1} 1/v$. After one round however, the behaviour changes, as the structure of the fire's shortest path to the barrier changes.

During the first round, the shortest fire path is always a straight line segment starting on the boundary of R_0 and orthogonal to it. After the first round completes however, the shortest fire path is instead a straight line segment starting at p_0. Therefore, the first round of the logarithmic spiral centered on the origin is followed by a short section of another logarithmic spiral with the same excentricity but centered around p_0, as illustrated in Figure 3.4.

This changes again, when the shortest fire path becomes tangential to the barrier curve at x_0, i. e. the tangent to the spiral curve at p_0 intersects the arc of the logarithmic spiral centered at p_0. From then on the shortest fire path always consists of a part of the barrier curve, followed by a short line segment tangential to it, as the fire first has to burn around the barrier curve constructed in

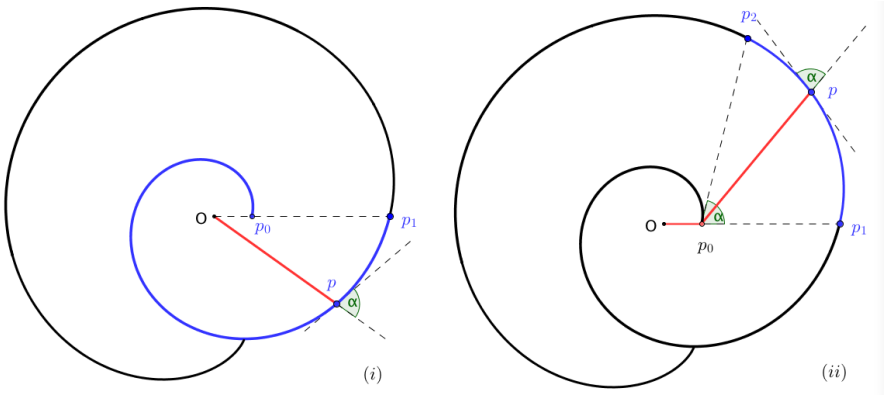


Figure 3.4: The barrier curve starts with two parts of logarithmic spirals of excentricity α, centered at 0 and p_0, respectively.

previous rounds. The barrier curve therefore forms a more complicated curve always dependent on the barrier curve in the previous rounds. However, as the building speed v is constant, the tangents of the curve always have angle α to the last straight line segment on the shortest fire path, as illustrated in Figure 3.5.

Dependent on the building speed v, the barrier curve constructed this way is either infinite or eventually closes in on itself. Klein et al. [57] show that the behaviour of this barrier curve FF_v is governed by a complex function $(e^{wZ} - sZ)^{-1}$, where w and s are real functions of v. Residue calculus for this function reveals that the curve contains the fire for any speed $v > v_c$ and provides an estimate for the number of rounds.

Klein et al. [57] also prove a stronger lower bound on the speed for a specific subclass of strategies, to which both FF_v and the piecewise logarithmic strategy by Bressan belong. This subclass of *spiralling* strategies consists of all barrier curves, that start on the boundary of the fire and visit the four coordinate half-axes in counter-clockwise order and at increasing distances from the origin. They show any spiralling strategy requires at least speed $v > 1.618\ldots$, the golden ratio, to contain a fire in the single barrier model.

3.2.2 The Angle Cover Model

Assume a fire starts at the origin $(0,0)$ and spreads along all outward rays at unit speed. This means, that fire does not burn around the corners of barriers as illustrated in contrast for the same barrier system in Figure 3.6. Any barrier either blocks an angle for the fire or has no effect, which means that no delaying barriers exist. This also means, that a barrier system containing the fire does not need to contain a connected enclosing barrier. Instead the fire is contained if there is a piece of barrier intersecting any outward ray starting in the fire origin in any direction. Hence we call this the angle cover model, as a barrier system has to cover every angle to contain the fire.

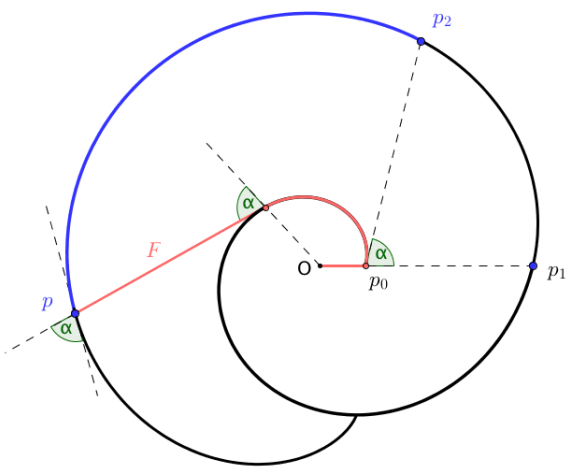


Figure 3.5: From point p_2 on the barrier curve results from wrapping around the barrier already constructed. The line segment F, the last piece of the shortest path from the fire source to the current barrier point p always has angle α to the barrier curve at that point. In addition, F simultaneously shrinks, as a bigger part of the shortest fire paths wraps around the previous curve, and grows, as the shortest fire path increases with time. The fighter will be successful if, and only if, F ever shrinks to zero.

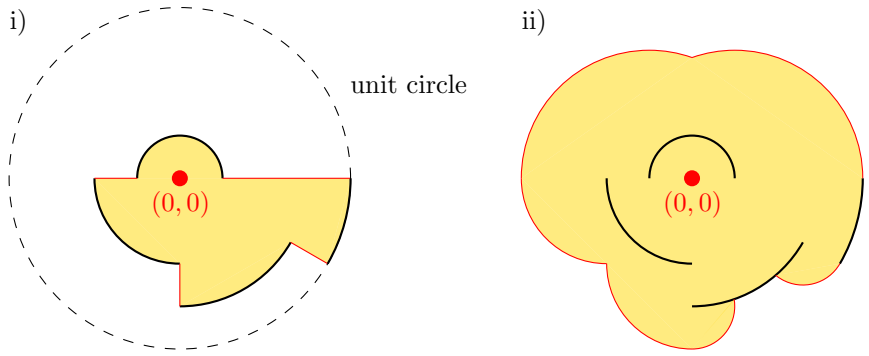


Figure 3.6: i) A barrier system for the angle cover problem for a speed of $v = \pi$.
ii) In the canonical model, the fire would escape from the same barrier system.

As this makes the fire strictly weaker than in the Bressan model, the upper bound of $v > 2$ also holds for the angle cover model. But exceeding that, we will show the following theorem.

Theorem 9
For any speed $v > 0$, it is possible to contain the fire in the angle cover model in the unit circle.

Proof. The harmonic series $\sum_{i=1}^{\infty} \frac{1}{i}$ is unbounded. Therefore for any value x there exists an n such that $\sum_{i=1}^{n} \frac{1}{i} \geq x$. Let n_v be the smallest such n for $x = \frac{2\pi}{v}$.

Consider the following strategy with all points given in polar coordinates:

Beginning in point $(\frac{1}{n_v}, 0)$ build a segment of a circle of length $\frac{v}{n_v}$. This segment can be build exactly in time $\frac{1}{n_v}$, and therefore the fire reaches the whole segment exactly at time of completion. Let us say this segment ends in point $(\frac{1}{n_v}, \alpha_1)$. Then continue building segments $i = 2$ to n_v of a circle of length $\frac{v}{n_v}$, always beginning in point $(\frac{i}{n_v}, \alpha_{i-1})$ and ending in $(\frac{i}{n_v}, \alpha_i)$. An example of this strategy for $v = \pi$ is illustrated in Figure 3.6.

Consider the projection of all those segments of a circle on the unit circle. Each segment i covers a segment of the unit circle of length $\frac{v}{n_v} \cdot \frac{n_v}{i} = \frac{v}{i}$. Moreover, by choice of start points, the projection of neighbouring segments do not overlap except for end and start points.

Therefore all segments together cover a segment of the unit circle of length

$$\sum_{i=1}^{n_v} \frac{v}{i} = v \cdot \sum_{i=1}^{n_v} \frac{1}{i} > 2\pi, \tag{3.1}$$

which covers the whole unit circle.

As all barriers are circular arcs with increasing radii around the fire origin and the last piece has exactly radius 1, all barrier pieces are built inside the unit circle, which concludes the proof. □

3.2.3 The Barrier Following Fire Model

In the barrier following fire model, the fire starts at the origin of the plane and has to be contained in the upper half-plane. To avoid that the fire instantaneously reaches the lower half-plane, we allow an arbitrarily small head-start barrier of length s in both directions along the x-axis. The fire spreads with unit speed, but only along any barrier already built.

The firefighter is restricted to building a horizontal barrier along the x-axis and vertical barriers attached to it and has to prevent the fire from ever reaching the end of the horizontal barrier. Hence the infinite horizontal barrier is the outer enclosing barrier that limits the fire to the upper half-plane, while the vertical barriers are delaying barriers. As the fire burns twice along each vertical barrier - first up and then down again - building as many of them as possible is crucial to effective strategies for this model. To avoid degenerate strategies that just build vertical barriers at infinitesimally short distances to each other, we require that a vertical barrier of height b has distance b to the next vertical barrier in the same direction.

Consider the following intuitive symmetric strategy. In both directions, the firefighters starts to build a vertical barrier at the end of the head-start barrier of length s at equal speed $\frac{v}{2}$. If $v < 2$ he is eventually reached by the fire at a point t_1. At this point he will have built two barriers of length $\frac{vt_1}{2}$ each. Hence, $\frac{vt_1}{2} + s = t_1 \Leftrightarrow t_1 = \frac{2s}{2-v}$ and the length of the first barrier is $b := \frac{vt_1}{2} = \frac{vs}{2-v}$.

Now he must build a part of the horizontal barrier in both directions before he is allowed to build the next vertical barrier, namely a piece of length b_1 again. He then starts the next vertical barrier until he is reached by the fire again, which will have length $\frac{2-2v}{v-2} b_1$. Following this approach repeatedly results in the i-th vertical barrier having length $b_i = \frac{2-2v}{v-2} b_{i-1} = \frac{2-2v}{v-2}^{i-1} b_1$. We will call this the growth factor $\beta_v = \frac{2-2v}{v-2}$. For $v = \frac{4}{3}$ it holds $\beta_v = 1$, so this strategy results in infinitely many barriers of equal length and distance as illustrated in Figure 3.7 .

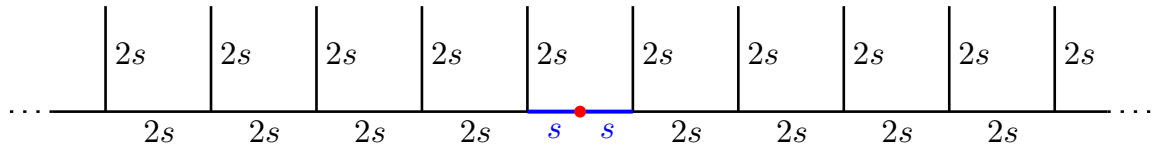


Figure 3.7: A infinite symmetric strategy for the barrier following fire model for $v = \frac{4}{3}$.

For $v > \frac{4}{3}$ it holds $\beta_v > 1$, so the length of the vertical barriers increases by a constant factor each time, e. g. the length of the vertical barriers doubles with each iteration for $\frac{3}{2}$. For $v < \frac{4}{3}$ it holds $\beta_v < 1$ so the length of the vertical barriers decreases by a constant factor with each iteration. This means hat the total length of all barriers built by this strategy is $\sum_{k=0}^{\infty} \beta_v^k 2 \frac{vs}{2-v}$. As $\sum_{k=0}^{\infty} x^k = \frac{1}{1-x}$ if $|x| < 1$, that sum is also bounded by $\frac{1}{1-\beta_v} 2 \frac{vs}{2-v}$, therefore the fire eventually reaches the end of the horizontal barrier if $v < \frac{4}{3}$.

Theorem 10

The infinite symmetric strategy contains the fire in the barrier following fire model if and only if $v \geq \frac{4}{3}$.

3.3 The Half-Plane Model

The main model, which we analyse in depth in this chapter, is the *half-plane model*. In the half-plane model, the fire spreads from the origin and continuously expands over time with unit speed according to the L_1 metric and has to be contained in the upper half-plane. As in Subsection 3.2.3, we prevent the fire from immediately spreading into the lower half-plane by allowing an arbitrarily small head-start barrier of length s into both directions along the x-axis. The goal is then to provide a barrier strategy that contains the fire in the upper half-plane.

The firefighter is restricted to barrier systems that consist of the horizontal enclosing barrier along the boundary of the half-plane and vertical delaying barriers attached to it. In contrast to strategies for the canonical continuous firefighting problem, these barrier system contain an infinite barrier as the enclosing barrier.

As established for the canonical model, we don't need to define the order in which the barriers in the system are built over time. Instead, we can analyse the order in which the fire reaches the barriers in a fixed barrier system. It is clear that when building a barrier system simultaneously to the fire spreading, every piece of barrier must be build before the fire reaches it. A barrier system can be built with a speed v if and only if the total length of barriers reached at time t never exceeds $v \cdot t$. The question we focus on is: What is the minimum speed v for which such a barrier system exists?

As our main results, we show that no such barrier system exists for a speed $v \leq 1.\overline{6}$ and provide a barrier system for a speed $v \geq 1.8772$.

3.3.1 Notation

To describe a strategy or barrier system in our model, we denote the i-th delaying barrier to the right by b_i. The part of the horizontal barrier between b_{i-1} and b_i is denoted by a_i. For simplicity of notation, we also refer to their length by a_i and b_i. If there is only a finite number r of vertical barriers in one direction, then a_{r+1} denotes the infinite piece of horizontal barrier after the last vertical barrier b_r. For the other direction, we use c_i and d_i respectively.

For convenience, $A_i := \sum_{j=1}^{i} a_j$ will denote the total length of horizontal barriers in the right direction up till and including a_i and $B_i := \sum_{j=1}^{i} b_j$ will denote the total length of vertical barriers in the right direction up till and including b_i. We define C_i and D_i equivalently for the left direction.

As the fire spreads over the barrier system, it represents a geodesic L_1 circle, which consumes the barriers when burning along them. The fire-front is the set of all points in the plane, which shortest non-barrier-crossing path to the fire origin has length t. We consider a point x on a barrier as *consumed* at time t if the fire has reached this point at time t. That means there exists a non-barrier-crossing path of length at most t from the fire origin to the point x. Hence, any piece of the barrier system is not consumed all at once, but as the fire burns along it. We call a point on a barrier, whose shortest non-barrier-crossing path to the fire has exactly length t, a *consumption point* at time t, so the consumption points are a subset of the fire front. We call the number of consumption points at time t the *current consumption* and a time interval with constant k consumption points at all times a *k-interval*.

We will usually consider k-intervals and current consumption separately for both directions. The fire front, consumption points, k-intervals and the effect of vertical delaying barriers are illustrated in Figure 3.8. After the fire reaches a delaying barrier for the first time, it may burn along the barrier system at multiple points. However, after reaching both ends and passing the top of a barrier there might be no consumption for a while as the delaying barrier has already been burned along from the other side.

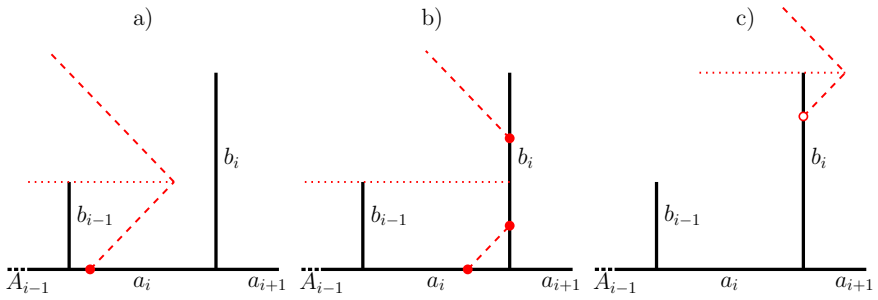


Figure 3.8: Fire spreading along delaying barriers. The dashed line shows the fire front at different times t, solid points represent consumption points, while empty points represent places, where the fire burns along the back of already consumed parts of the barrier b_i. In a) there is one consumption point, so there is a 1-interval in the right direction. In b) there are three consumption points and in c) there is a 0-interval in the right direction as there are no consumption points.

We define the *total consumption* $\mathscr{C}$ and *consumption-ratio* $\mathscr{Q}$ for a time interval $[t_1,t_2]$ in a barrier system:

$$\begin{aligned}\mathscr{C}(t_1,t_2) &:= \text{length of barrier pieces consumed by the fire between } t_1 \text{ and } t_2\\ \mathscr{Q}(t_1,t_2) &:= \tfrac{\mathscr{C}(t_1,t_2)}{t_2-t_1}.\end{aligned}$$

For the consumption in a time interval $[0,t]$, we will also write $\mathscr{C}(t)$ and $\mathscr{Q}(t)$ for short. In our setting, if $[t_0,t_1]$ is a k-interval, then $\mathscr{C}(t_1) = \mathscr{C}(t_0) + (t_1 - t_0)\cdot k$. As a consequence, $|\mathscr{Q}_{t_0} - k| > |\mathscr{Q}_{t_1} - k|$ so the consumption-ratio will always go towards k during a k-interval.

Note that all these definitions can easily be applied to either side of the barrier system separately to define the *directional consumption* and *directional consumption-ratio* in the left and right direction, denoted by $\mathscr{Q}^l(t)$, $\mathscr{Q}^r(t)$ and $\mathscr{C}^l(t)$, $\mathscr{C}^r(t)$ equivalently. Obviously, $\mathscr{Q}(t) = \mathscr{Q}^l(t) + \mathscr{Q}^r(t)$ and $\mathscr{C}(t) = \mathscr{C}^l(t) + \mathscr{C}^r(t)$.

As $\mathscr{C}(t)$ is exactly the sum of lengths of all barriers, it is necessary and sufficient to have $\mathscr{C}(t) \le v\cdot t$ to be able to build a barrier system with limited building speed v. This is equivalent to having $v \ge \sup_t \mathscr{Q}(t)$, so we will focus on bounding $\mathscr{Q}(t)$ at the local maxima to determine for which speed v a barrier system can be built.

3.4 Lower Bounds for the Half-Plane Model

We will start with some structural observations for optimal barrier systems, that lead to a straightforward initial lower bound of $v > 3/2 = 1.5$. This basic bound is based on the observation, that the consumption-ratio in both directions will repeatedly attain local maxima of at least 1 while local minima will still have a directional consumption-ratio of at least $1/2$. We will then refine the analysis of the consumption-ratio around local minima and maxima to obtain increasing lower bounds, culminating in a lower bound of $v > \frac{5}{3} = 1.\overline{6}$.

3.4.1 A Basic Bound

Observe, that a vertical barrier that is shorter than the predecessor in the same direction does not delay the fire. Hence, we can assume that vertical barriers in one direction increase strictly in length, so $b_i > b_{i-1}$ and $d_i > d_{i-1}$ for all $i > 1$. But we can show an even stronger bound on the growth of successive vertical barriers.

Lemma 10

Let S be a barrier system with $\mathscr{C}_S(t) \leq v \cdot t$ at all times t. Then there exists a barrier system S' with $\mathscr{C}_{S'}(t) \leq \mathscr{C}_S(t)$ at all times t, in which any vertical barrier b_i (or d_i) is more than twice as long as the previous barrier b_{i-1} (or d_{i-1}) in the same direction, i. e., we have $b_i > 2b_{i-1}$ and $d_i > 2d_{i-1}$ for all $i > 1$.

Proof. Assume S does not fulfil both properties $b_i > 2b_{i-1}$ and $d_i > 2d_{i-1}$. As argued above, we can at least assume $b_i > b_{i-1}$ and $d_i > d_{i-1}$ for all $i > 1$. Then we can construct a new barrier system S' with barriers denoted a' to d' with $b'_i > 2b'_{i-1}$ and $d'_i > 2d'_{i-1}$ while $\mathscr{C}_{S'}(t) \leq \mathscr{C}_S(t) \leq v \cdot t$ for all t.

The construction is identical for both directions, so w. l. o. g. we just consider the right direction. Let r be the number of barriers in S in the right direction, which can be infinity. Let b_k ($r \geq k > 1$) be the first vertical barrier in the right direction with $b_k \leq 2b_{k-1}$. Let $\Delta = b_k - b_{k-1}$.

To construct S', we remove b_k and move all following vertical barrier $2\Delta := 2(b_k - b_{k-1})$ to the right. If b_k is the last barrier in the right direction, nothing changes besides the removal of b_k. So, the right side of our barrier system S' consists of b'_i and a'_i precisely as follows:

$$\begin{aligned}
&\text{for } 1 \leq i < k && b'_i = b_i && a'_i = a_i && (3.2)\\
&\text{if } k < r && b'_k = b_{k+1} && a'_k = a_k + a_{k+1} + 2\Delta && (3.3)\\
&\text{for } k \leq i < r && b'_i = b_{i+1} && a'_i = a_{i+1} && (3.4)\\
&\text{if } r < \infty && && a'_r = a_{r+1}. && (3.5)
\end{aligned}$$

We will now give an injective map of any piece of barrier a'_i or b'_i in S' to a piece of barrier in S of at least the same length. As long as every piece of barrier in S' is consumed at the same time or later than its counterpart in S, we have $\mathscr{C}_{S'}(t) \leq \mathscr{C}_S(t)$. For $i < k$, we can clearly map a'_i to a_i and b'_i to b_i as S and S' are identical up to that point. Therefore, the consumption-ratio in the right direction is identical for S and S' until time $t = a_{k-1} + b_{k-1}$ when the fire reaches the top of b_{k-1} in S. We will consider the mapping for $i \geq k$ in more detail, beginning with a'_k.

a'_k can be mapped to a combination of a_k, a_{k+1} and b_k as illustrated in Figure 3.9. The piece of a'_k that's identical to a_k is consumed by the fire in S' at the exact same time and speed as in S. In S, the leftmost point of a_{k+1} is reached by the fire at time $a_k + b_{k-1} + 2\Delta$. Therefore the rightmost part of a'_k of the same length as a_{k+1} is reached and consumed at the by the fire in S' at the exact same time and speed as a_{k+1} in S. The same arguments hold, if b_k, was the last barrier in the right direction and a_{k+1} (and therefore a'_k) are infinite. The remaining part of a'_k of length 2Δ is at most as long as b_k,

as $2\Delta = 2(b_k - b_{k-1}) = b_k + (b_k - 2b_{k-1}) \geq b_k$, but first reached in S' at a time, when b_k has already been completely consumed in S. Every a'_i with $i > k$ can be mapped to the a_{i+1} of the same length as it is reached and therefore consumed exactly Δ later than its counterpart.

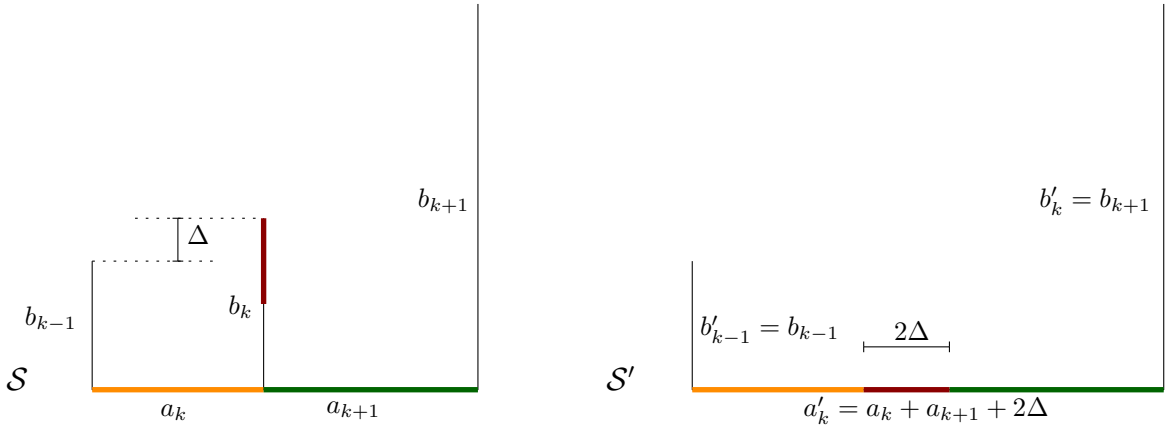


Figure 3.9: The consumption of the horizontal barrier a'_k in S can be mapped to a_k, a_{k-1} and b_k in S. The mapped pieces of equal length are indicated by color. Each piece of a'_k in S' is consumed at the same time or later than the corresponding barrier in S.

Assuming, b_k was not the last barrier in the right direction, b'_k can be mapped to b_{k+1}. While b'_k has the same length as b_{k+1}, a different point of it is reached first due to the removal of b_k. Every vertical barrier is first reached by the fire at the height of the previous vertical barrier. While b'_k is first reached at height b_{k-1}, b_{k+1} is first reached at height b_k, exactly Δ higher. b'_k is first reached at time $t + a_k + A_{k+1} + 2\Delta$, while b_{k+1} is reached at time $t + a_k + \Delta + a_{k+1}$, exactly Δ earlier. As b'_k is reached both later and at a lower point, every point on it is consumed later than the same point on b_{k+1}. Every b'_i with $i > k$ can be mapped to the b_{i+1} of the same length, which is reached at the same point, but exactly 2Δ earlier.

If S' still contains some barrier b'_j with $b'_j > 2b'_{j-1}$, this transformation can be repeated, which concludes the proof.

□

A similar observation can be made about the relation between length of the vertical delaying barriers and the length of the horizontal pieces in between.

Lemma 11

Let S be a barrier system with $\mathscr{C}_S(t) \leq v \cdot t$ at all times t. Then there exists a barrier system S' with $\mathscr{C}_{S'}(t) \leq \mathscr{C}_S(t)$ at all times t, in which any horizontal barrier a_i (or c_i) is at least as long as the previous vertical barrier b_{i-1} (or d_{i-1}) in the same direction, i. e., we have $a_i \geq b_{i-1}$ and $c_i \geq d_{i-1}$ for all $i > 1$.

Proof. By Lemma 10, we can assume S to fulfil $b_i > 2b_{i-1}$ and $d_i > 2d_{i-1}$ for all $i > 1$. Assume S does not fulfil $a_i \geq b_{i-1}$ and $c_i \geq d_{i-1}$ for all $i > 1$. Then we can construct a new barrier system S' with barriers denoted a' to d' with $a'_i \geq b'_{i-1}$ and $c'_i \geq d'_{i-1}$ for all $i > 1$, while $\mathscr{C}_{S'}(t) \leq \mathscr{C}_S(t) \leq v \cdot t$.

The construction is identical for both directions, so w. l. o. g. we just consider the right direction. Let r be the number of barriers in the right direction, which can be infinity. Let a_k ($r \geq k > 1$) be the first horizontal barrier in the right direction with $a_k < b_{k-1}$. Let $\Delta = \frac{1}{5}(b_{k-1} - a_k)$.

To construct S', we reduce the length of both b_{k-1} and b_k by 2Δ, move b_k by 3Δ and all following b_i with $i > k$ by 2Δ to the right, which changes the lengths of a_k and a_{k+1}. If b_k is the last barrier in the right direction, only b_k, b_{k-1} and a_k are affected, as $a_{k+1} = \infty$. So, the right side of our barrier system S' consists of b'_i and a'_i precisely as follows:

$$\text{for } 1 \leq i < k-1 \qquad b'_i = b_i \qquad a'_i = a_i \tag{3.6}$$

$$b'_{k-1} = b_{k-1} - 2\Delta \qquad a'_{k-1} = a_{k-1} \tag{3.7}$$

$$b'_k = b_k - 2\Delta \qquad a'_k = a_k + 3\Delta \tag{3.8}$$

$$\text{if } k+1 \leq r \qquad b'_{k+1} = b_k \qquad a'_k = a_{k+1} - \Delta \tag{3.9}$$

$$\text{for } k+1 < i \leq r \qquad b'_i = b_i \qquad a'_i = a_i \tag{3.10}$$

Similar to the proof of Lemma 10, we want to injectively map the consumption of every piece of barrier in S' to a piece in S that is consumed at the same time or later. For $1 \leq i < k-1$, we can map a'_i and b'_i in S' to the identical pieces in S, as the fire spreads through that part identically. Similarly, we can map every a'_i and b'_i for $i > k+1$ to their direct counterparts a_i and b_i in S. As the length of b'_{k+1} is identical to b_{k+1}, and the top of b'_{k+1} is reached exactly 2Δ later than the top of its counterpart b_{k+1} in S, all following b'_i and a'_i in S' are reached by the fire in the exact same manner, but 2Δ later than their direct counterparts in S.

Thus, we only need to consider the consumption of a'_{k-1}, b'_{k-1}, a'_k, b'_k, a'_{k+1} and b'_{k+1} in more detail. a'_{k-1} and b'_{k-1} are consumed identically to a_{k-1} and b_{k-1}, however, as b'_{k-1} is 2Δ shorter, there is a piece of b_{k-1} remaining that we can map the consumption of another piece to. b'_{k+1} is 2Δ further to the right than its counterpart b_{k+1}, but due to b'_k being exactly 2Δ shorter than b_k, b'_{k+1} is still first reached by the fire at the same time as b_{k+1}, albeit at a point 2Δ lower than in S. In addition, by our initial assumption $b'_{k+1} = b_{k+1} > 2b_k = 2(b'_k + 2\Delta) > 2b'_k$. Thus, b'_{k+1} is consumed in two phases - first a 2-interval of length b'_k, followed by a 1-interval of length $b'_{k+1} - 2b'_k$. This is clearly slower than its counterpart b_{k+1} in S, which is consumed first in a 2-interval of length $b_k = b'_k + 2\Delta$, followed by a 1-interval of length $b_{k+1} - 2b_k = b'_{k+1} - 2b'_k + 4\Delta$. Thus, we can map, b'_{k+1} to b_{k+1}.

The combination of the remaining barriers, a'_k, b'_k and a'_{k+1} can be mapped to their counterparts a_k, b_k and a_{k+1} as well as the remaining free piece of b_{k-1} of length 2Δ, but not one to one due to the differences in length. Instead of describing the mapping piece by piece, we will compare the exact order in which those barriers are consumed in S and in S' and the resulting consumption, as illustrated in Figure 3.10. As long as the consumption of the pieces in S' never exceeds those of the pieces in S,

there exists the injective mapping we require. We will start at the time t_0, when the fire reaches the top of b'_{k-1}. This is the first moment, when the consumption of S and S' differs due to the remaining piece of b_k in S being consumed.

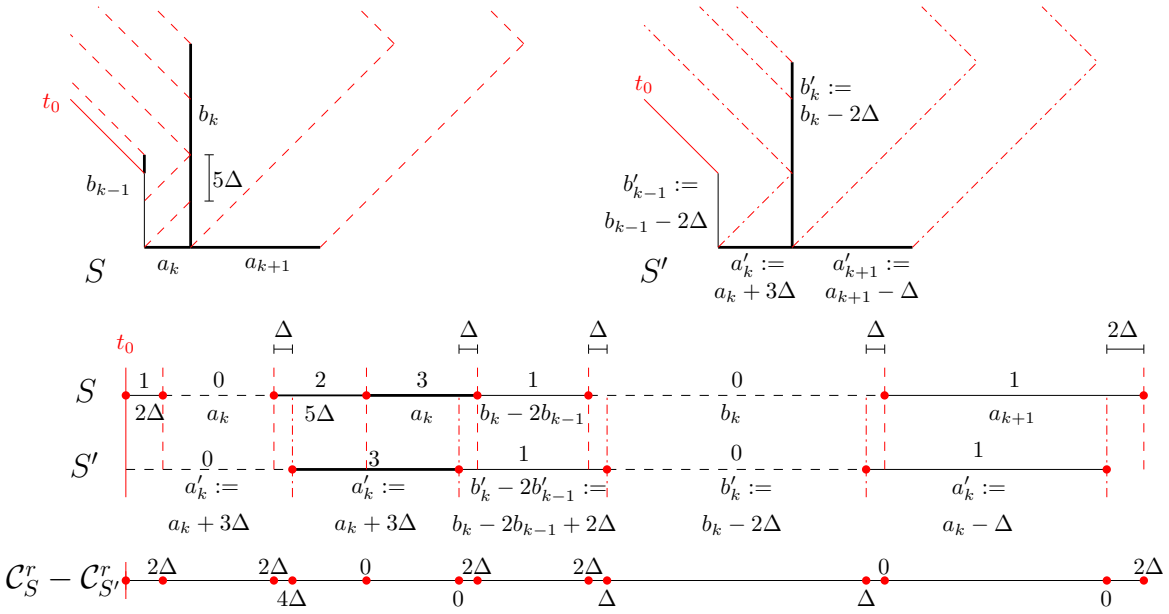


Figure 3.10: The combined consumption of the horizontal barriers a'_k, b'_k, a'_{k+1} in S happens later than that of their counterparts a_k, b_k and a_{k+1} as well as the remaining free piece of b_{k-1}. The resulting consumption intervals and the difference in consumption at each interval end is given below the figures illustrating the construction. The red dashed and dotted lines indicate changes in consumption in either barrier system.

Thus, the consumption of the aforementioned pieces in S begins with a 1-interval of length 2Δ, until the remaining piece of b_{k-1} is consumed. This is followed by a 0-interval of length a_k, until the fire reaches the vertical barrier b_k at height b_{k-1}. As the fire burns along b_k both up and down, this creates a 2-interval of length $b_k - a_{k+1} = 5\Delta$, until the fire reaches the beginning of the horizontal barrier a_k. The added consumption of a_k results in a 3-interval of length a_k. The remainder of b_k is then consumed in a 1-interval of length $b_k - 2b_{k-1}$. After the fire burns down the back of b_k without consumption, resulting in a 0-interval of length b_k, the final piece a_{k+1} is then consumed in a single 1-interval of length a_k.

The consumption of a'_k, b'_k and a'_{k+1} is a little simpler. Due to the length adjustments, $a'_k = b'_{k-1}$ as $a'_k = a_k + 3\Delta = a_k + \frac{3}{5}(b_{k-1} - a_k) = \frac{3}{5}b_{k-1} + \frac{2}{5}a_k = b_{k-1} - \frac{2}{5}(b_{k-1} - a_k) = b_{k-1} - 2\Delta = b'_{k-1}$. In addition, $b'_k = b_k - 2\Delta \geq 2b_{k-1} - 2\Delta > 2b'_{k-1}$ by construction. Thus, we start with a 0-interval of length $a'_k = b_{k-1} = a_k + 3\Delta$ until the fire reaches both b'_k and a'_k simultaneously, which is followed by a 3-interval of the same length. The remainder of b'_k is then consumed in a single 1-interval of length $b'_k - 2b'_{k-1} = b_k - 2b_{k-1} + 2\Delta$. After the fire burns down the back of b'_k without consumption,

resulting in a 0-interval of length $b'_k = b_k - 2\Delta$, the final piece a'_{k+1} is then consumed in a single 1-interval of length $a'_k = a_k - \Delta$.

Note, that when considering all of S and S', the consumption of b_{k+1} and b'_{k+1} might start at some point during these time intervals. But we only need to consider a_k, b_k and a_{k+1} as well as the remaining free piece b_{k-1} of length 2Δ, as we already mapped the consumption of all other pieces in S.

Now, if the consumption of the considered intervals in S' exceeds the one in S at any point, it will also do so at the end of one of the intervals due to the linearity of the consumption within one interval. But, as illustrated in Figure 3.10, the consumption in S lies below S' at every interval change. Hence, there exists an injective mapping as required.

By construction, $a'_k = b'_{k-1}$. Thus, repeated application of the transformation above result in a strategy S' fulfilling $a_i \geq b_i$ for all i.

Note that our proof about the lower consumption of S' requires S to fulfil $b_i > 2b_{i-1}$ for all i, but one step of our construction might destroy this property as the length of b_{k-1} gets reduced. In such a case, we apply the transformation from the proof of Lemma 10, so S' fulfils $b_i > 2b_{i-1}$ before each construction step. This can only lengthen any a_i and thus, does not affect the overall construction. □

This means that when given an arbitrary barrier system containing the fire for some $v < 2$, we can assume the following conditions to hold:

$$a_{i+1} \geq b_i \quad \text{and} \quad b_{i+1} \geq 2b_i \quad \forall i \geq 1, \tag{3.11}$$

$$\text{similarly} \quad c_{i+1} \geq d_i \quad \text{and} \quad d_{i+1} \geq 2d_i \quad \forall i \geq 1. \tag{3.12}$$

If a barrier systems fulfils these conditions, the order of consumption of vertical and horizontal barriers in a single direction is very regular due to the following observation: When the fire reaches the top of a vertical barrier b_i at some time t_i (compare Figure 3.12), every barrier a_k and b_k with $k \leq i$ has been completely consumed, as for every point on a_k or b_k the shortest non-barrier-crossing path has length smaller than $A_i + b_i = t_i$.

Hence, a 0-interval in the right direction will begin at such times t_i and $\mathscr{C}^r(t_i) = A_i + B_i - s$, where s denotes the length of the head-start not contributing to the consumption. Moreover, for a single direction this results in a repeating sequence of k-intervals of specific lengths and k as illustrated in Figure 3.11. Due to the $a_{i+1} > b_i$, the 0-interval starting at t will have length exactly b_i. We say, that the barrier b_i *generated* that 0-interval. This observation holds equivalently for both directions.

We will utilize this structural property both for the construction of viable strategies as well as in our proofs for lower bounds for v. As an initial result, we will derive a basic lower bound on the necessary speed v to contain the fire in the half-plane model, by considering the consumption in one direction at the beginning and at the end of every 0-interval.

Theorem 11: Basic Lower Bound

In the half-plane model, the fire can not be contained with speed $v \leq 1.5$.

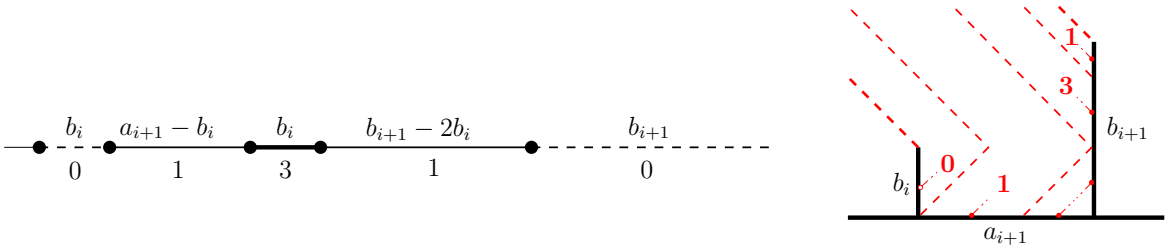


Figure 3.11: A sequence of k-intervals for the right direction. The length is given above each interval and the current consumption below.

Proof of Theorem 11

For a barrier system to contain the fire in the half-plane model, we must have $\mathscr{C}(t) \leq v \cdot t$ at all times t.

Lemma 12

Let S be a barrier system in the half-plane model with $\mathscr{C}(t) \leq 2 \cdot t$ at all times t. Then, after some fixed threshold time t_t, the consumption-ratio in one direction always exceeds 1 at the beginning of 0-intervals in that direction. Moreover, for any arbitrarily large time t there exist 0-intervals that begin after t in both directions.

Proof. As without vertical delaying barriers, the consumption-ratio just goes towards 2, S has an unbounded number of vertical barriers in at least one direction. W. l. o. g. assume this is the right direction. By Lemma 10, we can also assume $b_i > 2b_{i-1}$ and $d_i > 2d_{i-1}$ for all $i > 1$ in S. Therefore, for any constant, there is some i large enough, such that b_i and all subsequent barriers b_j with $j > i$ exceed that constant. In particular, this holds for $2s$, e. g. $b_i > 2s$ for all i large enough.

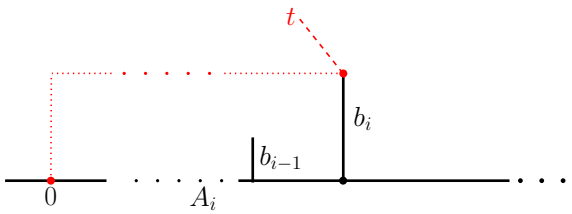


Figure 3.12: At some time $t = A_i + b_i$ the fire will reach the top of a vertical barrier b_i.

Consider a moment when the fire reaches the end of some barrier b_i as illustrated in Figure 3.12. This happens at time $t = b_i + A_i$ and Lemma 10 implies we have $\mathscr{C}^r(t) = A_i + B_i - s$.

$$\begin{aligned} \mathscr{C}^r(t) &= A_i + B_i - s = A_i + b_i + B_{i-1} - s && \mid B_{i-1} > 2s \text{ for } i \text{ large enough} \\ &> A_i + b_i + s > t + s > t \end{aligned} \tag{3.13}$$

Hence, when the fire reaches the top of a vertical barrier (i. e. the beginning of a 0-interval) in the right direction at times t large enough, $\mathscr{Q}^r(t) > 1$. Therefore, S also has repeated 0-intervals in the left direction as well. Otherwise $\mathscr{Q}^l(t)$ would go towards 1 and $\mathscr{Q}(t) > 2$ at such times.

Hence the same argument holds for the other direction as well, which concludes the proof. □

Lemma 12 implies that both directions repeatedly attain local maxima where the directional consumption-ratio exceeds 1. To derive a lower bound, we also need to consider the local minima.

Lemma 13

Let S be a barrier system for the half-plane model with $\mathscr{C}(t) \leq 2 \cdot t$ at all times t. Then, after some fixed threshold time t_t, the consumption-ratio in one direction always exceeds $\frac{1}{2}$ in both directions.

Proof. By Lemma 12, the directional consumption-ratio in either direction repeatedly exceeds 1. As S contains the fire for some speed $v < 2$, this implies that the consumption-ratio in each direction also repeatedly falls below 1, such that the sum of consumption-ratio in both directions stays below 2 at all times.

W. l. o. g. consider a local minimum of the directional consumption in the right direction at a time t_{min} with $\mathscr{Q}^r(t_{min}) < 1$. During a k-interval in one direction, the consumption-ratio in that direction goes towards k. As the consumption-ratio either increases or decreases the whole time during an interval, a local minimum (or maximum) will always lie at the end of an interval and will have consumption-ratio greater (smaller for maxima) than k. Therefore, t_{min} must lie at the end of a 0-interval.

Consider any moment, when we reach the end of a 0-interval in the right direction after the fire has burned along the back of some barrier b_i. This happens at time $t = A_i + 2b_i$, when the fire reaches the horizontal barrier and starts consuming a_{i+1}. As before, Lemma 10 implies $\mathscr{C}^r(t) = A_i + B_i - s$.

$$\begin{aligned} \mathscr{C}^r(t_{min}) &= A_i + B_i - s = A_i + b_i + B_{i-1} - s \qquad \mid B_{i-1} > s \text{ for } i \text{ large enough} \\ &> A_i + b_i = \frac{1}{2}(2A_i + 2b_i) > \frac{1}{2}(A_i + 2b_i) \geq \frac{1}{2} t_{min} \end{aligned} \tag{3.14}$$

Hence, for times t large enough, at the end of all 0-intervals and therefore at all local minima with $\mathscr{Q}^r < 1$, $\mathscr{Q}^r(t) > \frac{1}{2}$. This holds equivalently for the left direction, which concludes the proof. □

At a time t_i, when the fire reaches the top of a vertical barrier b_i in the right direction, we have $\mathscr{Q}^r(t) > 1$ by Lemma 12 and $\mathscr{Q}^l(t) > \frac{1}{2}$ by Lemma 13. Therefore, $\mathscr{Q}(t) > \frac{3}{2} = 1.5$, which concludes the proof of Theorem 11.

3.4.2 A Lower Bound Based on Helpful Cycles

Let S be a barrier system that contains the fire in the half-plane model for some speed $v = \frac{3}{2} + \varepsilon < 2$. By Lemma 12, we know that S has infinitely many 0-intervals in both directions. We therefore structure the consumption intervals into cycles based on these 0-intervals.

Let z_i denote the i-th 0-interval in the right direction. Between two consecutive 0-intervals z_i and z_{i+1}, there is a regular sequence of k-intervals with $k \geq 1$ as presented in Subsection 3.4.1. However, the exact lengths of these k-intervals depends on the barrier system. Without any further assumptions on the structure of the barrier system, we refer to this whole sequence of k-intervals by w_i. Together we refer to such a pair (z_i, w_i) as the cycle i. For simplicity, let w_i and z_i also denote the length of these intervals. For the left direction, we denote the cycles by (y_i, v_i) instead.

Let $t_i = \sum_{j=1}^{i-1}(w_j + z_j)$ denote the start time of z_i, i. e. the beginning of cycle i. Let $w_{i,k}$ be the total length of all k-intervals within w_i and K_i be the largest k within w_i with $w_{i,k} > 0$. Then, let $W_i^{\mu} = \sum_{k=1}^{K_i} k \cdot w_{i,k}$ denote the total consumption during w_i. This is equal to the consumption $\mathscr{C}^r(t_i, t_{i+1})$ during the whole cycle i as there is no consumption during the 0-interval z_i.

Like in Subsection 3.4.1 we will use the interaction of high and low values of the consumption-ratio in either direction to obtain a stronger lower bound. By Lemma 12, the directional consumption-ratio at the beginning of a 0-interval, i. e., at the end of a cycle, exceeds 1. To keep the global consumption-ratio below $\frac{3}{2} + \varepsilon$, this must be countered by a consumption-ratio below $\frac{1}{2} + \varepsilon$ in the other direction. Let a *helpful cycle* be a cycle during which a cycle (y_j, w_j) from the other direction ends. We will show that the end of a cycle must lie inside an interval with $\frac{1}{2} < \mathscr{Q}^r(t) \leq \frac{1}{2} + \varepsilon$ around the local minimum of a helpful cycle. Bounding both the size of this interval and the consumption-ratio within a single helpful cycle, we will obtain the following theorem:

Theorem 12
In the half-plane model, the fire can not be contained with speed $v \leq 1.53069$.

Proof of Theorem 12

We will start with some structural observations. The first cycle is somewhat unique, as its 0-interval is generated by the head-start barrier instead of a vertical delaying barrier. But we can derive the following fundamental properties for all following cycles (z_i, w_i) for $i > 1$.

Lemma 14
i) $W_i^{\mu} \geq z_i + z_{i+1}$
ii) $z_i > w_i$.

Proof. Let b_i be the delaying barrier generating the 0-interval z_i. By Lemma 11 and Lemma 10, we can assume the k-intervals within a cycle to be regular as illustrated in Figure 3.11.

i) The next 0-interval z_{i+1} will be generated by b_{i+1} with $k > i$, so $z_{i+1} = b_{i+1}$. However, both a_{i+1} and b_{i+1} are consumed completely during (z_i, w_i), so $W_i^{\mu} = a_{i+1} + b_{i+1} \geq z_i + z_{i+1}$.

ii) This follows directly form the regular k-intervals, as w_i contains a 3-interval of length $b_i = z_i$ as well as a 1-interval of length $b_{i+1} - 2b_i > 0$.

□

Now consider a helpful cycle (z_i, w_i) in the right direction, during which a cycle (y_h, v_h) from the other direction ends at some time t.

Lemma 15

For helpful cycles i, we have

$$z_i \geq \frac{1/2 - \varepsilon}{1/2 + \varepsilon} \sum_{j=1}^{i-1} (w_j + z_j). \tag{3.15}$$

Proof. To counter $\mathscr{Q}^l(t) > 1$ at the end of v_h, we must have $\mathscr{Q}^r(t) < \frac{1}{2}$. However, by the regular structure of the k-intervals, $\mathscr{Q}^r(t) \leq \frac{1}{2} + \varepsilon$ only in an interval around the unique local minimum of (z_i, w_i), which happens at the end of the z_i at $t_i + z_i$. Thus, we must have $\mathscr{Q}^r(t_i + z_i) < 1/2 + \varepsilon$, hence

$$\mathscr{Q}(t_i + z_i) = \frac{\sum_{j=1}^{i-1} W_j^{\mu}}{z_i + t_i} \leq 1/2 + \varepsilon \quad \Leftrightarrow \quad \sum_{j=1}^{i-1} W_j^{\mu} \leq (z_i + t_i)(1/2 + \varepsilon). \tag{3.16}$$

By Lemma 12, we know that the directional consumption-ratio exceeds 1 at t_i:

$$\mathscr{Q}(t_i) = \frac{\sum_{j=1}^{i-1} W_j^{\mu}}{t_i} \geq 1 \quad \Leftrightarrow \quad \sum_{j=1}^{i-1} W_j^{\mu} \geq t_i. \tag{3.17}$$

Combining Equation 3.16 and Equation 3.17, we get:

$$\begin{aligned}
& t_i \leq (z_i + t_i)(1/2 + \varepsilon) \\
\Leftrightarrow \quad & t_i (1/2 - \varepsilon) \leq z_i (1/2 + \varepsilon) \\
\Leftrightarrow \quad & \sum_{j=1}^{i-1} (w_j + z_j) \leq \frac{1/2 + \varepsilon}{1/2 - \varepsilon} \cdot z_i \\
\Leftrightarrow \quad & z_i \geq \frac{1/2 - \varepsilon}{1/2 + \varepsilon} \sum_{j=1}^{i-1} (w_j + z_j)
\end{aligned} \tag{3.18}$$

□

By these arguments and Lemma 13, the directional consumption-ratio minimum of a helpful cycle i at $t_i + z_i$ attains a value between $1/2 + \varepsilon$ and $1/2$. To keep the global consumption-ratio below $\frac{3}{2} + \varepsilon$, the end of the cycle in the other direction must lie in the interval around that minimum, for which the consumption-ratio is below $1/2 + \varepsilon$ as illustrated in Figure 3.13.

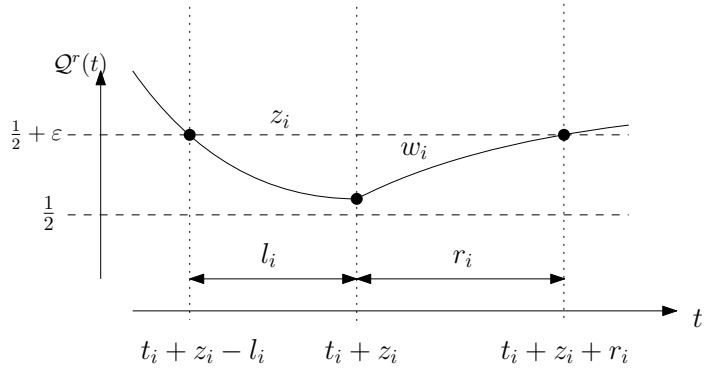


Figure 3.13: A local minimum in a helpful cycle.

Lemma 16

Let $[t_i+z_i-l_i, t_i+z_i+r_i]$ be the interval around the minimum at t_i+z_i of a helpful cycle, where the consumption-ratio in that direction is below or equal to $1/2+\varepsilon$. Then

$$l_i \leq \frac{\varepsilon}{1/2+\varepsilon}(t_i+z_i) \leq \frac{\varepsilon}{(1/2-\varepsilon)(1/2+\varepsilon)}\, z_i \tag{3.19}$$

and

$$r_i \leq \frac{\varepsilon}{1/2-\varepsilon}(t_i+z_i) \leq \frac{\varepsilon}{(1/2-\varepsilon)^2}\, z_i. \tag{3.20}$$

Proof. By Lemma 13, we know that $\mathscr{C}^r(t_i+z_i) \geq 1/2(t_i+z_i)$. There also is at least consumption 1 between t_i+z_i and $t_i+z_i+r_i$, as it is part of w_i, hence:

$$1/2+\varepsilon = \frac{\mathscr{C}^r(t_i+z_i+r_i)}{t_i+z_i+r_i} \geq \frac{\mathscr{C}^r(t_i+z_i)+r_i}{t_i+z_i+r_i} \geq \frac{\frac{1}{2}\cdot(t_i+z_i)+r_i}{t_i+z_i+r_i}. \tag{3.21}$$

Solving this inequality for r_i yields

$$r_i \leq \frac{\varepsilon}{1/2-\varepsilon}(t_i+z_i) = \frac{\varepsilon}{1/2-\varepsilon}\left(\sum_{j=1}^{i-1}(z_j+w_j)+z_i\right). \tag{3.22}$$

By Lemma 15, r_i can be related directly to z_i:

$$r_i \leq \frac{\varepsilon}{1/2-\varepsilon}\left(\frac{1/2+\varepsilon}{1/2-\varepsilon}z_i+z_i\right) = \frac{\varepsilon}{(1/2-\varepsilon)^2}\, z_i. \tag{3.23}$$

Equivalently, applying the fact, that there is consumption 0 between $t_i+z_i-l_i$ and t_i+z_i, yields the bound for l_i:

$$l_i \leq \frac{\varepsilon}{1/2+\varepsilon}(t_i+z_i) = \frac{\varepsilon}{1/2-\varepsilon}\left(\sum_{j=1}^{i-1}(z_j+w_j)+z_i\right) \leq \frac{\varepsilon}{(1/2-\varepsilon)(1/2+\varepsilon)}\, z_i. \tag{3.24}$$

□

From Lemma 13 it follows that the consumption-ratio in one direction can attain values of at most $1+\varepsilon$. But that does not mean that the consumption-ratio of a single cycle $\frac{\mathscr{C}(t_i,t_{i+1})}{z_i+w_i}$ is at most $1+\varepsilon$. However, the consumption-ratio of a cycle can not be arbitrarily large, as the consumption-ratio is greater than 1 at the beginning of the cycle and at most $1+\varepsilon$ at the end. This observation can be used to obtain an upper bound on z_{i+1}:

Lemma 17
For helpful cycles (z_i,w_i), we have

$$z_{i+1} \leq \left(1+\frac{3-2\varepsilon}{1-2\varepsilon}\,\varepsilon\right) w_i.$$

Proof. Assume an arbitrary cycle has a consumption-ratio of $1+\beta\varepsilon$, so:

$$\frac{\mathscr{C}(t_i,t_{i+1})}{z_i+w_i} = 1+\beta\varepsilon \quad \Leftrightarrow \quad \mathscr{C}(t_i,t_{i+1}) = (z_i+w_i)(1+\beta\varepsilon) \tag{3.25}$$

Now consider the consumption-ratio at the end of the cycle:

$$\begin{aligned} & \frac{\mathscr{C}(t_{i+1})}{t_{i+1}} = \frac{\mathscr{C}(t_i)+\mathscr{C}(t_i,t_{i+1})}{t_i+z_i+w_i} \leq 1+\varepsilon \\ \Leftrightarrow \quad & \mathscr{C}(t_i)+\mathscr{C}(t_i,t_{i+1}) \leq (t_i+z_i+w_i)(1+\varepsilon). \end{aligned} \tag{3.26}$$

By Lemma 12, $\mathscr{C}(t_i) > t_i$. Applying this and Equation 3.25 to the left side of Equation 3.26 yields:

$$\begin{aligned} & t_i+(z_i+w_i)(1+\beta\varepsilon) \leq (t_i+z_i+w_i)(1+\varepsilon) \\ \Leftrightarrow \quad & t_i+(z_i+w_i)+(z_i+w_i)\beta\varepsilon \leq t_i+(z_i+w_i)+t_i\varepsilon+(z_i+w_i)\varepsilon \\ \Leftrightarrow \quad & (z_i+w_i)\beta\varepsilon \leq t_i\varepsilon+(z_i+w_i)\varepsilon \\ \Leftrightarrow \quad & (z_i+w_i)(\beta-1) \leq t_i = \sum_{j=1}^{i-1}(w_j+z_j) \end{aligned} \tag{3.27}$$

For helpful cycles, we can apply Lemma 15 on the right side and Lemma 14 ii) on the left side of Equation 3.27 and solve for β:

$$\begin{aligned} & 2z_i(\beta-1) \leq \frac{1/2+\varepsilon}{1/2-\varepsilon}\cdot z_i \\ \Leftrightarrow \quad & \beta-1 \leq \frac{1/2+\varepsilon}{1-2\varepsilon}\cdot \\ \Leftrightarrow \quad & \beta \leq \frac{3/2-\varepsilon}{1-2\varepsilon} \end{aligned} \tag{3.28}$$

So for all helpful cycles the consumption-ratio within the cycle does not exceed $1+\beta\varepsilon$, i.e.

$$\frac{W_i^\mu}{w_i+z_i} \leq 1+\beta\varepsilon \quad \Leftrightarrow \quad W_i^\mu \leq (1+\beta\varepsilon)(w_i+z_i)$$

Applying Lemma 14 i) to the left and, in a second step, Lemma 14 ii) to the right side results in

$$\begin{aligned} & z_i + z_{i+1} \leq z_i + \beta\varepsilon z_i + (1+\beta\varepsilon w_i) \\ \Leftrightarrow \quad & z_{i+1} \leq (1+2\beta) w_i \\ \Leftrightarrow \quad & z_{i+1} \leq \left(1 + \frac{3-2\varepsilon}{1-2\varepsilon}\,\varepsilon\right) w_i. \end{aligned} \tag{3.29}$$

□

Now consider a z_{i+1} from a helpful cycle as depicted in Figure 3.14. By Lemma 17, we can bound

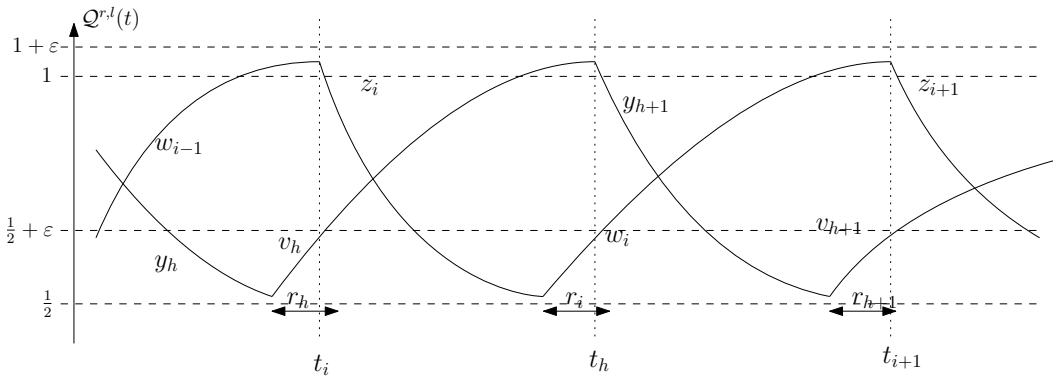


Figure 3.14: Interweaving of directional consumption of left and right cycles.

z_{i+1} in terms of w_i. In turn, we want to bound w_i in terms of y_{h+1}. As established in the proof of Lemma 16, the cycle endpoint at t_{i+1} with it consumption ≥ 1 can lie at most r_{h+1} to the right of the minimum at $t_h + y_{h+1}$ to keep to total consumption-ratio below $(\frac{3}{2}+\varepsilon)$. Equivalently, the minimum at $t_i + z_i$ can only lie r_i to the left of t_{h+1}, as seen in Figure 3.14. Hence

$$w_i \leq y_{h+1} + r_i + r_{h+1} \tag{3.30}$$

Applying Lemma 16 results in

$$w_i \leq y_{h+1} + \frac{\varepsilon}{1/2-\varepsilon}(t_i+z_i) + \frac{\varepsilon}{1/2-\varepsilon}(t_{h+1}+y_{h+1}). \tag{3.31}$$

Note, that the r_i is only relevant if $t_i + z_i \leq t_{h+1}$, so we can assume as much. Then applying Lemma 15 and Lemma 16 we obtain

$$\begin{aligned} w_i &\leq y_{h+1} + \tfrac{\varepsilon}{1/2-\varepsilon}(t_{h+1}) + \tfrac{\varepsilon}{1/2-\varepsilon}(t_{h+1} + y_{h+1}) \\ &\leq y_{j+1}\left(1 + \tfrac{\varepsilon}{(1/2-\varepsilon)^2} + \tfrac{\varepsilon(1/2+\varepsilon)}{(1/2-\varepsilon)^2}\right). \end{aligned} \tag{3.32}$$

Lemma 17 and Lemma 16 can be applied to derive equivalent bounds for y_{h+1}, in terms of v_j and v_j in terms of z_i:

$$y_{h+1} \leq \left(1 + \frac{3-2\varepsilon}{1-2\varepsilon}\,\varepsilon\right) v_j \quad \text{and} \quad v_j \leq z_i \left(1 + \frac{\varepsilon}{(1/2-\varepsilon)^2} + \frac{\varepsilon\,(1/2+\varepsilon)}{(1/2-\varepsilon)^2}\right). \tag{3.33}$$

Combining all arguments from Lemma 17, Inequality 3.32 and Inequality 3.33 to bound z_{i+1} directly in terms of z_i and ε yields

$$z_{i+1} \leq \left(1 + \frac{3-2\varepsilon}{1-2\varepsilon}\,\varepsilon\right)^2 \left(1 + \frac{\varepsilon}{(1/2-\varepsilon)^2} + \frac{\varepsilon\,(1/2+\varepsilon)}{(1/2-\varepsilon)^2}\right)^2 z_i. \tag{3.34}$$

On the other hand, Lemma 15 gives a bound in the other direction, when combined with Lemma 14 ii):

$$z_{i+1} \geq \frac{1/2-\varepsilon}{1/2+\varepsilon} \sum_{j=1}^{i} (w_j + z_j) \geq \frac{1/2-\varepsilon}{1/2+\varepsilon}(w_i + z_i) \geq \frac{1-2\varepsilon}{1/2+\varepsilon}\, z_i. \tag{3.35}$$

Satisfying both Inequality 3.34 and Inequality 3.35 yields a contradiction unless $\varepsilon \geq 0.03069$.

Note that our construction implicitly assumed that all cycles are helpful, and thus, helpful cycles are consecutive. However, if this is not the case, then both ends of a not helpful cycle (z_i, w_i) and thus, the complete cycle must lie within the interval with $\mathscr{D}^l(t) < \frac{1}{2} + \varepsilon$ around a local minimum of a helpful cycle in the other directions, which only results in an even stronger bound.

This concludes the proof of Theorem 12.

3.4.3 A Lower Bound Based on Intervals with Bounded Directional Consumption-Ratio

For the basic bound derived in Subsection 3.4.1 one only needed to consider a single point of consumption in one direction and combine it with the minimum in directional consumption in the other direction. In Subsection 3.4.2, we refined this idea by focusing on the interval around a helpful minimum in directional consumption-ratio, which has low enough consumption-ratio to counter high values in the directional consumption-ratio in the other direction. We will now extend on this idea by splitting the whole timeline for each direction into time intervals with directional consumption-ratio either above or below $\frac{v}{2}$. Obviously, intervals with directional consumption-ratio above $\frac{v}{2}$ in one

direction must be completely contained within an interval with directional consumption-ratio below $\frac{v}{2}$ in the other direction. By finding bounds for the size of these intervals, we will proof the following stronger lower bound.

Theorem 13

In the half-plane model, the fire can not be contained with speed $v \leq 1.625$.

Proof of Theorem 13

Assume there exists a barrier system for a speed $v = 1.5 + \varepsilon$, so $\sup_t \mathscr{Q}(t) < 3/2 + \varepsilon$. By Lemma 10 and Lemma 11, we can assume $b_i > 2b_{i-1}$ and $a_i > b_{i-1}$ for all $i > 1$.

Consider a time t_r, at which the consumption in the right direction reaches a maximum between two 0-intervals, for which by Lemma 12 $\mathscr{Q}^r(t_r) > 1$. By Lemma 13, the consumption in the left direction is always at least $\frac{1}{2}$, hence $\mathscr{Q}^r(t_r) \leq 1 + \varepsilon$ and equivalently for the other direction. Consequently, it must regularly hold $\mathscr{Q}_l(t) \leq 1/2 + \varepsilon$ at local minima in both directions.

Now consider the situation at a local minimum t_{min}, w.l.o.g. in the right direction. Again, as shown in the proof of Lemma 13, a local minimum is reached at the end of a 0-interval. Therefore

$$\begin{aligned} \frac{\mathscr{C}_{t_{min}}}{t_{min}} \leq \frac{A_i + B_i}{A_i + 2\,b_i} &\leq 1/2 + \varepsilon \\ (1/2 - \varepsilon) A_i &\leq (1 + 2\varepsilon)\, b_i - B_i \\ A_i &\leq \frac{(2 + 4\varepsilon)}{1 - 2\varepsilon}\, b_i - \frac{2}{1 - 2\varepsilon} B_i. \end{aligned} \tag{3.36}$$

However, the maximum in consumption-ratio between two 0-intervals is not reached a the beginning of the 0-interval. By Lemma 12, we know that the consumption-ratio exceeds 1 at the beginning of a 0-interval. By the regularity of the k-intervals, the 0-interval generated by a vertical barrier b_{i+1} is preceded by a 1-interval of length $b_{i+1} - 2b_i$. But during a k-interval, the directional consumption goes towards k, and can therefore only have a consumption-ratio above k at the end if it the consumption-ratio decreased during the whole interval. This means the directional consumption-ratio decreased towards 1 during this interval to arrive at a value above 1 at the end of it. Hence, the directional consumption-ratio at the beginning of this 1-interval must be even higher than at the beginning of the 0-interval.

However, to keep the combined consumption-ratio below $\frac{3}{2} + \varepsilon$, a local maximum in directional consumption consumption-ratio can not exceed exceed $1 + \varepsilon$ by Lemma 13. Therefore, the directional consumption-ratio increased during the 3-interval before the 1-interval. Thus, the directional consumption-ratio has a local maximum at t_{max} at the end of the 3-interval, when the fire has consumed

length $2b_i$ of barrier b_{i+1}. Hence,

$$\begin{aligned}
\frac{\mathscr{C}_{t_{max}}}{t_{max}} &\leq 1+\varepsilon \\
\frac{A_i+B_i+a_{i+1}+2b_i}{A_i+a_{i+1}+2\,b_i} &\leq 1+\varepsilon \\
B_i &\leq \varepsilon\, A_i+\varepsilon a_{i+1}+2\varepsilon b_i.
\end{aligned}$$

Here, we can directly apply Equation 3.36 to obtain a lower bound for a_{i+1}:

$$\begin{aligned}
B_i &\leq \frac{\varepsilon\,(2+4\varepsilon)}{1-2\varepsilon}\,b_i-\frac{2\varepsilon}{1-2\varepsilon}B_i+\varepsilon a_{i+1}+2\varepsilon b_i \\
\left(1+\frac{2\varepsilon}{1-2\varepsilon}\right)B_i &\leq \left(2\varepsilon+\frac{\varepsilon\,(2+4\varepsilon)}{1-2\varepsilon}\right)b_i+\varepsilon\, a_{i+1} \\
\Rightarrow \qquad \left(1+\frac{2\varepsilon}{1-2\varepsilon}\right)b_i &< \left(2\varepsilon+\frac{\varepsilon\,(2+4\varepsilon)}{1-2\varepsilon}\right)b_i+\varepsilon\, a_{i+1} \\
\varepsilon\, a_{i+1} &\geq \left(1+\frac{2\varepsilon}{1-2\varepsilon}-2\varepsilon-\frac{\varepsilon\,(2+4\varepsilon)}{1-2\varepsilon}\right)b_i \\
&\geq \left(\frac{1-2\varepsilon}{1-2\varepsilon}+\frac{2\varepsilon}{1-2\varepsilon}+\frac{4\varepsilon^2-2\varepsilon}{1-2\varepsilon}+\frac{-4\varepsilon^2-2\varepsilon}{1-2\varepsilon}\right)b_i \\
&\geq \frac{1-4\varepsilon}{1-2\varepsilon}\,b_i \\
a_{i+1} &\geq \frac{1-4\varepsilon}{\varepsilon\,(1-2\varepsilon)}\,b_i=\frac{1-4\varepsilon}{\varepsilon-2\varepsilon^2}
\end{aligned} \tag{3.37}$$

To shorten this in future formulae, we define $\alpha \coloneqq \frac{1-4\varepsilon}{\varepsilon(1-2\varepsilon)}$.

Now we can use Equation 3.36 to obtain a lower bound on b_i by applying Equation 3.37 to every summand in $A_i=\sum_{j=1}^{i} a_j$.

$$\begin{aligned}
\frac{1-4\varepsilon}{\varepsilon\,(1-2\varepsilon)}B_{i-1} &\leq \frac{2+4\varepsilon}{1-2\varepsilon}\,b_i-\frac{2}{1-2\varepsilon}B_i \\
\frac{1-4\varepsilon}{\varepsilon\,(1-2\varepsilon)}B_{i-1}+\frac{2}{1-2\varepsilon}b_{i-1} &\leq \frac{(2+4\varepsilon)}{1-2\varepsilon}\,b_i-\frac{2}{1-2\varepsilon}\,b_i \\
\left(\frac{1-4\varepsilon}{\varepsilon\,(1-2\varepsilon)}+\frac{2\varepsilon}{\varepsilon\,(1-2\varepsilon)}\right)B_{i-1} &\leq \frac{4\varepsilon}{1-2\varepsilon}\,b_i \\
\frac{1}{\varepsilon}B_{i-1} &\leq \frac{2\varepsilon}{1-2\varepsilon}\,b_i \\
\Leftrightarrow \qquad b_i &\geq \frac{1-2\varepsilon}{4\varepsilon^2}B_{i-1} \\
\Rightarrow \qquad b_i &\geq \frac{1-2\varepsilon}{4\varepsilon^2}b_{i-1}=\left(\frac{1}{(2\varepsilon)^2}-\frac{1}{2\varepsilon}\right)b_{i-1}
\end{aligned} \tag{3.38}$$

Again, for future use, we define $\beta \coloneqq \frac{1-2\varepsilon}{4\varepsilon^2}$.

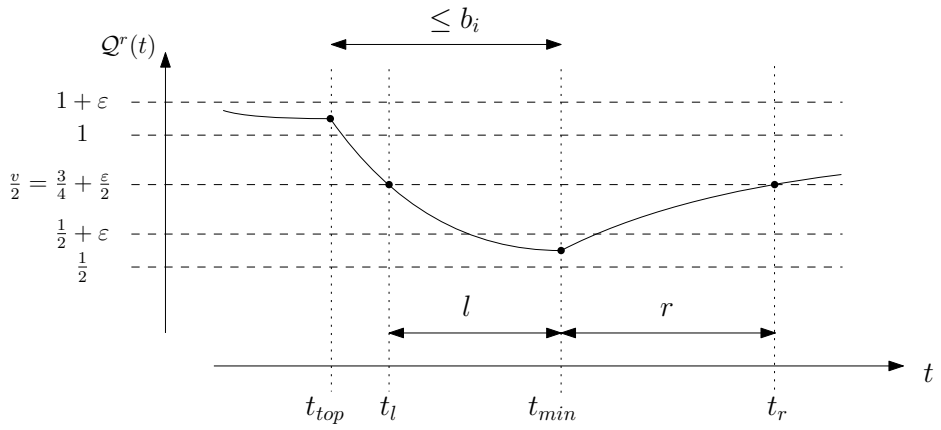


Figure 3.15: The consumption-ratio $\mathscr{Q}^r(t)$ around t_{min}.

For $\varepsilon \leq \frac{1}{2}$, Equation 3.38 yields $b_i \geq 0$, for $\varepsilon \leq 1/4$, this yields $b_i \geq 2b_{i-1}$, and for ε approaching 0, the factor approaches infinity. Similarly, Equation 3.37 yields $a_{i+1} \geq 0$ for $0 < \varepsilon \leq 1/4$, $a_{i+1} > b_i$ for $\varepsilon \leq 1/4\left(5-\sqrt{17}\right) \approx 0.22$, and $a_i \to \infty$ for $\varepsilon \to 0$. As we already know $b_i \geq 2b_{i-1}$ and $a_{i+1} > b_i$ by the much simpler arguments given in Subsection 3.4.1, this bound gives us new information about strategies for speeds $v = 3/2 + \varepsilon \leq 1.72$. Thus, from now on, we will assume $\varepsilon < 0.22$.

Let t_{min} be a time, when the consumption-ratio in the right direction attains a local minimum. Similar to Subsection 3.4.2, we will now use the bounds obtained above to determine the maximum size of the time interval around t_{min}, for which $\mathscr{Q}^r(t) \leq v/2 = 3/4 + \varepsilon/2$. Similar to the notion of helpful cycles in the proof of Theorem 12, we will call that interval the *helpful interval* around t_{min}. We can assume w. l. o. g., that $\mathscr{Q}^l$ attains a local maximum with $\mathscr{Q}^l > 1$ at a time t_{max} sometime during that helpful interval. We call the interval around t_{max} with $\mathscr{Q}^l(t) \geq v/2 = 3/4 + \varepsilon/2$ the *costly interval* around t_{max}. This costly interval must lie completely in the helpful interval around t_{min}, otherwise the combined consumption-ratio of both directions exceeds v. We use bounds on the size of these intervals to create a contradiction for some ε, which will lead to an improved lower bound on v.

Consider the consumption-ratio $\mathscr{Q}^r(t)$ around t_{min} as illustrated in Figure 3.15. As it is a local minimum, t_{min} must mark the end of a 0-interval after the fire has moved down the back of some barrier b_i. Let t_{top} be the time when the fire reaches the top of b_i at the beginning of the 0-interval. By Lemma 12, we known that the directional consumption-ratio $\mathscr{Q}^r(t)$ exceeds one at such moments. Hence

$$\mathscr{Q}^r(t_{top}) = \frac{\mathscr{C}^r(t_{top})}{t_{top}} > 1 \quad \Rightarrow \quad \mathscr{C}^r(t_{top}) > t_{top} = A_i + b_i > b_i. \tag{3.39}$$

Let $[t_l, t_r]$ be the interval around t_{min} with $\mathscr{Q}^r(t) = 3/4 + \varepsilon/2$, i.e. $\mathscr{Q}^r(t_l) = 3/4 + \varepsilon/2 = \mathscr{Q}^r(t_r)$. As we assume a local maximum t_{max} with $\mathscr{Q}^l(t_{max}) > 1$ in the other direction to lie in this interval, we know that $\mathscr{Q}^r(t_{min})$ must not exceed $1/2 + \varepsilon$ to counter this maximum and keep $\mathscr{Q}(t)(t_{max})$ below v. Therefore

$$\mathscr{Q}^r(t_{min}) = \frac{\mathscr{C}^r(t_{min})}{t_{min}} \leq 1/2 + \varepsilon \quad \Rightarrow \quad \mathscr{C}^r(t_{min}) \leq (1/2 + \varepsilon)\, t_{min}. \tag{3.40}$$

The 0-interval is exactly b_i long, hence $t_{min} = t_{top} + b_i$. As there is no consumption during a 0-interval, we get $\mathscr{C}^r(t_{min}) = \mathscr{C}^r(t_{top}) > t_{top}$. Applying this to Inequality 3.40 yields

$$\begin{aligned} & t_{top} < \mathscr{C}^r(t_{top}) \leq (1/2+\varepsilon)\, t_{top} + (1/2+\varepsilon)\, b_i \\ \Rightarrow \quad & (1/2-\varepsilon)\, t_{top} < (1/2+\varepsilon)\, b_i \\ \Rightarrow \quad & b_i > \frac{1-2\varepsilon}{1+2\varepsilon} t_{top} \end{aligned} \tag{3.41}$$

Now lets consider the endpoints of the helpful interval $[t_l, t_r]$. By definition,

$$\mathscr{Q}^r(t_l) = \frac{\mathscr{C}^r(t_l)}{t_l} = 3/4 + \varepsilon/2 \quad \Rightarrow \quad \mathscr{C}^r(t_l) = (1/2+\varepsilon)\, t_l. \qquad \text{| equivalent for } t_r \tag{3.42}$$

Again, as there is no consumption during a 0-interval, we get $\mathscr{C}^r(t_l) = \mathscr{C}^r(t_{top}) > t_{top}$. Let l be the length of the left part of the helpful interval relative to b_i, i. e. $lb_i \coloneqq t_{min} - t_l$. By this definition, $t_l = t_{min} - lb_i = t_{top} + b_i - lb_i$. Combining both these observations with Equation 3.42, we get

$$\begin{aligned} \Rightarrow \quad & t_{top} < \mathscr{C}^r(t_{top}) \leq (3/4+\varepsilon/2)\, (t_{top} + (1-l)\, b_i) \\ \Rightarrow \quad & (1/4-\varepsilon/2)\, t_{top} < (3/4+\varepsilon/2)\, (1-l)\, b_i \end{aligned} \tag{3.43}$$

By construction $t_{top} = A_i + b_i > b_i$. In addition, as we assume $\varepsilon < 0.22$, $(1/4-\varepsilon/2) > 0$, and thus, $(1/4-\varepsilon/2)\, t_{top} > (1/4-\varepsilon/2)\, b_i$. Applying this to Equation 3.43 eliminates t_{top} giving

$$\begin{aligned} & (1/4-\varepsilon/2)\, b_i < (3/4+\varepsilon/2)\, (1-l)\, b_i \\ \Rightarrow \quad & (3/4+\varepsilon/2)\, l < 1/2+\varepsilon \\ \Rightarrow \quad & l < \frac{2+4\varepsilon}{3+2\varepsilon} \end{aligned} \tag{3.44}$$

We similarly define $rb_i \coloneqq t_r - t_{min} \leq t_{top} + b_i$ and examine $t_r = t_{min} + rb_i = t_{top} + b_i + rb_i$. We know that the consumption-ratio in the right direction between t_{min} and t_r is at least 1 the whole time, hence

$$\mathscr{C}^r(t_r) \leq \mathscr{C}^r(t_{min}) + rb_i = \mathscr{C}^r(t_{top}) + rb_i > t_{top} + rb_i. \tag{3.45}$$

We can again combine the definition of r above, Inequality 3.45 and Equation 3.42 to attain

$$\begin{aligned} & t_{top} + rb_i < \mathscr{C}^r(t_{top}) + rb_i \leq (3/4+\varepsilon/2)\, (t_{top} + (1+r)\, b_i) \\ \Rightarrow \quad & (1/4-\varepsilon/2)\, t_{top} < (3/4+\varepsilon/2)\, b_i - (1/4+\varepsilon/2)\, rb_i \\ \Rightarrow \quad & (1/4-\varepsilon/2)\, b_i < (3/4+\varepsilon/2)\, b_i - (1/4+\varepsilon/2)\, rb_i \\ \Rightarrow \quad & (1/4+\varepsilon/2)\, r < (1/2+\varepsilon) \\ \Rightarrow \quad & r < \frac{2+4\varepsilon}{1-2\varepsilon} \end{aligned} \tag{3.46}$$

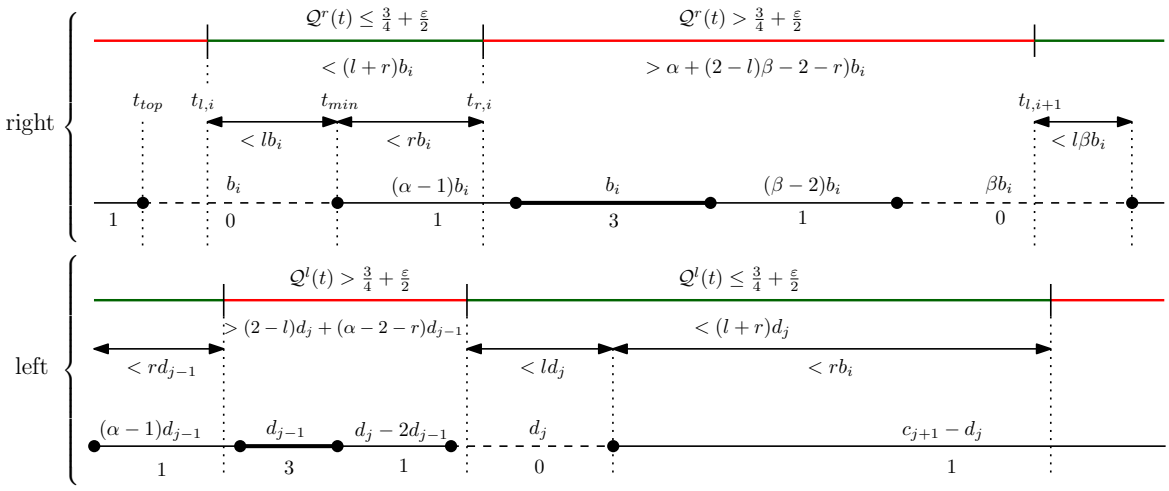


Figure 3.16: Construction of the size of the helpful (green) and costly (red) intervals around the local minimum t_{min} in both directions. The helpful interval in the left direction must completely cover the costly interval in the other direction and vice versa.

Of course, this holds for all b_i. Now we can relate the size of the interval $[t_l, t_r]$ to the interval between one local minimum and the next.

As shown in Subsection 3.4.1, we can assume $a_i > b_{i-1}$ and $b_i > 2b_{i-1}$, such that the sequence and lengths of the intervals after t_{top} are fix as illustrated in Figure 3.16: t_{top} is the beginning of a 0-interval of length b_i, followed by a 1-interval of length $a_{i+1} - b_i > (\alpha-1)\,b_i$. At the end of the following 3-interval of length b_i, $\mathcal{Q}^r$ attains a local maximum, and after a further 1-interval of length $b_{i+1} - 2b_i > (\beta-2)\,b_i$, the fire has reached the top of the next vertical barrier of length b_{i+1}. The cycle then begins anew with a 0-interval of length $b_{i+1} > \beta b_i$.

This means we can determine a lower bound on the size of the interval between $t_{r,i}$ and $t_{l,i+1}$, for which $\mathcal{Q}^r > 3/4 + \varepsilon/2$. As $t_{r,i} = t_{min,i} + rb_i$ and $t_{l,i+1} = t_{min,i} + (\alpha-1)\,b_i + b_i + (\beta-2)\,b_i + (1-l)\,b_{i-1} > t_{min,i} + (\alpha+(2-l)\beta-2)\,b_i$, this costly interval around the local maximum generated at the barrier b_{i+1} has size at least $(\alpha+(2-l)\beta-2-r)\,b_i$.

This means, that in the left direction, there must be a barrier d_j, such that the helpful interval around the local minimum generated by that barrier has at least the same length. This gives a lower bound on d_j:

$$\begin{aligned}(l+r)d_j &\geq (\alpha+(2-l)\beta-2-r)\,b_i \\ d_j &\geq \frac{\alpha+(2-l)\beta-2-r}{l+r}\,b_i \\ d_j &> \frac{-(1+\varepsilon)(16\varepsilon^2+2\varepsilon-1)}{8x^2(1+2\varepsilon)}\,b_i \end{aligned} \tag{3.47}$$

But the costly interval around local maximum generated at that d_j must fit inside $[t_{l,i}, t_{r,i}]$. This gives an upper bound on d_j.

$$\begin{aligned}(l+r)b_i &\geq (1-l)d_j + (d_j - 2d_{j-1}) + d_{j-1} + (\alpha - 1 - r)\, d_{j-1} \\ &\geq (2-l)\, d_j + (\alpha - r - 2) d_{j-1} \\ &\geq (2-l)\, d_j \qquad \text{if } \alpha \geq r+2 \\ d_j &\leq \frac{l+r}{2-l}\, b_i < \frac{2+4\varepsilon}{1-2\varepsilon}\, b_i \end{aligned} \tag{3.48}$$

$\alpha \geq r+2$ holds exactly if $0 < \varepsilon \leq 0.125$. However, $\frac{(1-\varepsilon)(16\varepsilon^2+2\varepsilon-1)}{8x^2(1+2\varepsilon)} < \frac{2+4\varepsilon}{1-2\varepsilon}$ for $0 < \varepsilon < 0.1273$. This is a contradiction to the fact, that a helpful interval around a local minimum in the one direction must cover the costly interval around the local maximum in the other direction, which concludes the proof of Theorem 13.

3.4.4 A Lower Bound of $v > \frac{5}{3}$

The best known lower bound can be shown by examining the exact moment in detail, when the fire reaches the top of a vertical barrier in one direction. The consumption-ratio at that time can then be combined with the consumption-ratio at the next local maxima in the other direction to prove the following theorem:

Theorem 14
In the half-plane model, the fire can not be contained with speed $v \leq 5/3 = 1.\overline{6}$.

Proof of Theorem 14

Assume there exists a barrier system S consisting of horizontal barriers along the x-axis and vertical barriers attached to it. Further assume for S that $\mathscr{C}(t) \leq v \cdot t$ at all times t for some $v = (1+V)$ with $V \leq \frac{2}{3}$. For this we will construct a contradiction by identifying a specific time t_S at a local maximum, for which $\mathscr{C}(t_S) > (1+V) \cdot t_S$.

By Lemma 10 and Lemma 11, we can assume $b_i > 2b_{i-1}$ and $d_i > 2d_{i-1}$, as well as $a_i > b_{i-1}$ and $c_i > d_{i-1}$ for all $i > 1$. Consider a moment when the fire reaches the top of some barrier b_i at time $t = A_i + b_i$ as illustrated in Figure 3.17. By Lemma 13, we have for large enough i

$$\mathscr{C}^r(t) > t + s > t \tag{3.49}$$

We now consider the situation in the left direction at the same time $t = b_i + A_i$. Let d_j denote the last vertical barrier, whose upper end was reached by the fire, so $t = d_j + C_j + \delta$ with $0 \leq \delta < c_{j+1} + d_{j+1} - d_j$.

W. l. o. g. we assume that $b_{i+1} + A_{i+1} \geq d_{j+1} + C_{j+1}$. Otherwise, there must be multiple vertical barriers in the right direction whose upper ends are reached by the fire after it reaches the upper end

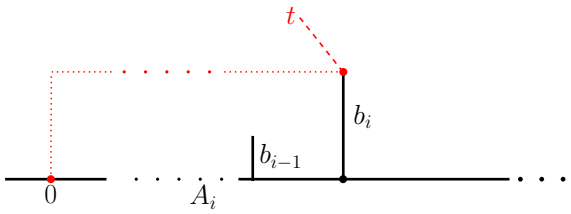


Figure 3.17: At some time $t = A_i + b_i$ the fire will reach the top of a vertical barrier b_i.

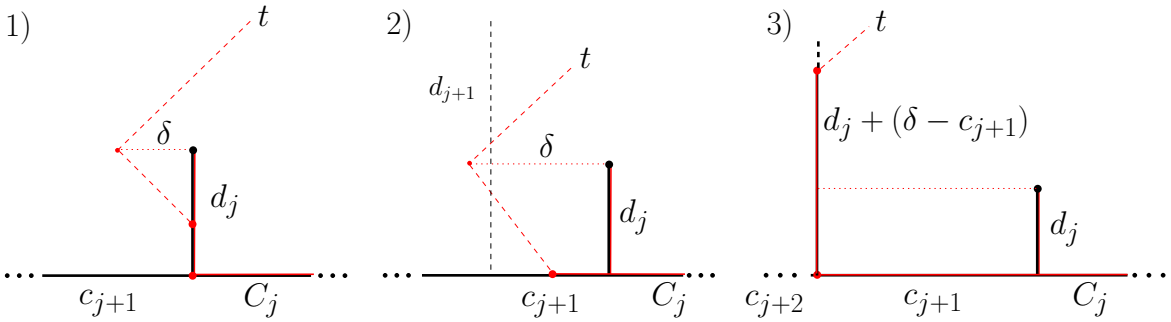


Figure 3.18: All three possible situations for the left side to be in at time t. Note that in case 2) the fire might have reached d_{j+1}, which does not affect our considerations.

of d_j and before it reaches the upper end of d_{j+1}. In that case, we can assume that b_i is the last among those, such that $b_{i+1} + A_{i+1} \geq d_{j+1} + C_{j+1}$ holds.

We split our consideration in three cases based on the consumption of c_{j+1} as illustrated in Figure 3.18:

1. $0 \leq \delta < d_j$
2. $d_j \leq \delta < d_j + c_{j+1}$
3. $d_j + c_{j+1} \leq \delta < c_{j+1} + d_{j+1} - d_j$

In the first case, the fire has not reached the horizontal barrier c_{j+1} yet after passing over d_j; in the second case, it has reached c_{j+1}, but not its end; in the third case the fire has completely consumed c_{j+1}.

In Case 3, $\delta = d_j + c_{j+1} + \lambda$ for some $0 \leq \lambda < d_{j+1} - 2d_j$. Then

$$\begin{aligned} \mathscr{C}^l(t) &\geq C_{j+1} + D_j + 2d_j + \lambda - s \\ &> (d_j + C_j) + (d_j + c_{j+1}) + \lambda \\ &= t \end{aligned} \tag{3.50}$$

Inequality (3.50) together with Inequality (3.49) already gives $\mathscr{C}(t) > 2t > (1+V) \cdot t$, which is a contradiction.

For both remaining cases, we will derive a lower bound for d_j. We will then consider the moment $t_1 = 2d_j + C_{j+1}$, when the fire reaches the end of the horizontal barrier c_{j+1}, i. e. a local maximum in consumption-ratio in the left direction. Using the lower bound on d_j, we will prove $\mathscr{C}(t_1) > (1+V)\cdot t_1$.

Case 1: $0 \leq \delta < d_j$

In Case 1, $\mathscr{C}^l(t) > C_j + D_j - s = C_j + d_j + D_{j-1} - s > C_j + d_j$, since $D_{j-1} > s$ for j large enough. Now at time t, it must hold:

$$\begin{aligned} \mathscr{C}(t) = \mathscr{C}^r(t) + \mathscr{C}^l(t) &< (1+V)\cdot t && \mid \text{Inequality (3.49)} \\ \Rightarrow \quad C_j + d_j &< V(d_j + C_j + \delta) \\ \Rightarrow \quad (V-1)C_j &> (1-V)d_j - V\delta && \mid (V < 1) \\ \Leftrightarrow \quad C_j &< {}^{V}\!/_{(1-V)}\cdot\delta - d_j \end{aligned} \tag{3.51}$$

$V \leq {}^2\!/_3$ implies $\frac{V}{(1-V)} \leq 2$ by direct calculation, which gives bounds for C_j, d_j:

$$\begin{aligned} C_j &< 2\delta - d_j < d_j && \mid \delta < d_j \text{ in Case 1} \\ \Rightarrow \quad 2d_j &> C_j + \delta \\ \Leftrightarrow \quad d_j &> {}^1\!/_2(C_j + \delta) \end{aligned} \tag{3.52}$$

Case 2: $d_j \leq \delta < d_j + c_{j+1}$

In Case 2 a part of c_{j+1} of length $(\delta - d_j)$ has already been consumed, so $\mathscr{C}^l(t) \geq D_j + C_j + (\delta - d_j) - s > d_j + C_j + (\delta - d_j) = C_j + \delta$, as $D_{j-1} > s$ for j large enough. Now at time t it must hold

$$\begin{aligned} \mathscr{C}(t) = \mathscr{C}^r(t) + \mathscr{C}^l(t) &< (1+V)\cdot t && \mid \text{Inequality (3.49)} \\ \Rightarrow \quad C_j + \delta &< V(d_j + C_j + \delta) \\ \Rightarrow \quad (1-V)(C_j + \delta) &< V d_j \\ \Rightarrow \quad d_j &> \frac{(1-V)}{V}(C_j + \delta) \end{aligned} \tag{3.53}$$

$V \leq {}^2\!/_3$ implies $\frac{(1-V)}{V} \geq {}^1\!/_2$ by direct calculation, which gives the bound:

$$d_j > {}^1\!/_2(C_j + \delta) \tag{3.54}$$

This is the same bound as found for Case 1 in Inequality (3.52).

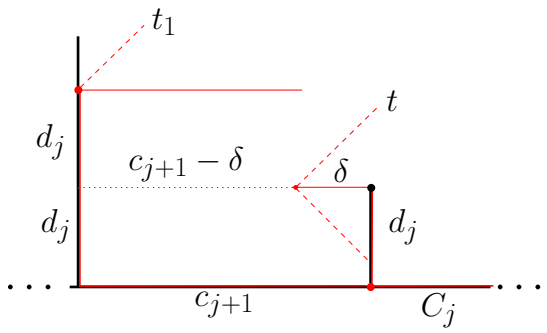


Figure 3.19: After $d_j + c_{j+1} - \delta$ additional time after t, the fire has reached the end of c_{j+1} and has also consumed a piece of length $2d_j$ of the next vertical barrier.

Deriving the contradiction $\mathscr{C}(t_1) > (1+V)\cdot t_1$

Now we consider time $t_1 = C_{j+1} + 2d_j > t$, when the fire reaches the end of the horizontal barrier c_{j+1}. As for any time, at time t_1, it must hold

$$\begin{aligned}
& & \mathscr{C}(t_1) &= \mathscr{C}^r(t_1) + \mathscr{C}^l(t_1) \leq (1+V)\cdot t_1 \\
\Leftrightarrow & & \mathscr{C}^l(t_1) &\leq (1+V)\cdot t_1 - \mathscr{C}^r(t_1) \\
& & &= (1+V)\cdot t + (1+V)(t_1 - t) - (\mathscr{C}^r(t) + \mathscr{C}^r(t,t_1)) \\
& & &\leq Vt + (1+V)(t_1 - t) + t - \mathscr{C}^r(t) \quad | \text{ Ineq. (3.49)} && (3.55) \\
\Rightarrow & \mathscr{C}^l(t_1) + s & &< Vt + (1+V)(t_1 - t) && (3.56)
\end{aligned}$$

By construction, $t_1 = C_{j+1} + 2d_j$. As $t = d_j + C_j + \delta$, this means $t_1 = t + (d_j + c_{j+1} - \delta)$. Due to Lemma 10, we know that the fire has not reached the end of d_{j+1} yet, hence $\mathscr{C}^l(t_1) \geq 3d_j + C_{j+1} - s$. Hence, we arrive at the following inequalities:

$$\begin{aligned}
& 3d_j + C_{j+1} &&< V(d_j + C_j + \delta) + (1+V)(d_j + c_{j+1} - \delta) \\
\Leftrightarrow \quad & -V(c_{j+1} - \delta) &&< (V-1)\delta + (V-1)C_j + (2V-2)d_j \qquad | (1 > V) \\
\Leftrightarrow \quad & c_{j+1} - \delta &&> \frac{1-V}{V}\delta + \frac{1-V}{V}C_j + 2\frac{1-V}{V}d_j . && (3.57)
\end{aligned}$$

$V \leq 2/3$ implies $\frac{(1-V)}{V} \geq 1/2$ by direct calculation, which gives the bound:

$$\begin{aligned}
& c_{j+1} - \delta > 1/2\delta + 1/2 C_j + d_j \\
\Leftrightarrow \quad & d_j + c_{j+1} - \delta > 1/2\delta + 1/2 C_j + 2d_j && (3.58)
\end{aligned}$$

Now in both cases we got $d_j > 1/2(C_j + \delta)$ (Inequalities (3.52) and (3.54)), so we can apply that and conclude:

$$t_1 - t = d_j + c_{j+1} - \delta > C_j + d_j + \delta = t = A_i + b_i \tag{3.59}$$

So we know, that in both cases $t_1 - t > b_i + A_i$. Now consider the situation in the right direction again (compare Figure 3.17). At $t + b_i$ the fire reaches the horizontal barrier a_{i+1} behind b_i. Additionally, by assumption $b_{i+1} + A_{i+1} \geq d_{j+1} + C_{j+1}$, the fire has not reached the top of the next barrier b_{i+1} at t_1. This means, that between $t + b_i$ and t_1, there is always at least consumption 1 in the right direction, which means the fire has consumed barriers of length at least A_i, hence $\mathscr{C}^r(t, t_1) \geq A_i$.

As our whole consideration is based on inequalities, we will consider a limiting case with a contradiction that can be extended to our given barrier system S. More precisely, assume, that Inequality (3.55) is tight for some t_1^*, so:

$$\begin{aligned} & \mathscr{C}^l(t_1^*) = Vt + (1+V)(t_1^* - t) + t - \mathscr{C}^r(t) \\ \Leftrightarrow \quad & \mathscr{C}^r(t) + \mathscr{C}^l(t_1^*) = (1+V)t_1^* \end{aligned} \tag{3.60}$$

By our arguments above, $\mathscr{C}^r(t, t_1^*) \geq A_i$ and hence

$$\mathscr{C}(t_1^*) = \mathscr{C}^r(t, t_1^*) + \mathscr{C}^r(t) + \mathscr{C}^l(t_1^*) \geq (1+V)t_1^* + A_i > (1+V)t_1^*, \tag{3.61}$$

which is a contradiction for this limiting case.

Now in our given barrier system S we have $t_1 = t_1^* + x$ for some $x > 0$. As everything except c_{j+1} is fixed at t, this additional time results in additional consumption of at least horizontal barriers of length x in both directions in comparison to the edge case. Hence we can extend the contradiction:

$$\begin{aligned} \mathscr{C}(t_1) &= \mathscr{C}^l(t_1) + \mathscr{C}^r(t) + \mathscr{C}^r(t, t_1) \\ &= \mathscr{C}^l(t_1^*) + \mathscr{C}^r(t) + \mathscr{C}^r(t, t_1^*) + 2x \\ &= (1+V)t_1^* + 2x + A_i \\ &> (1+V)(t_1^* + x) = (1+V)t_1. \end{aligned} \tag{3.62}$$

This concludes the proof of Theorem 14.

3.5 Upper bounds

3.5.1 A Recursive Strategy for $v \geq \frac{17}{9}$

Motivated by Lemma 10 and Lemma 11 shown in Subsection 3.4.1, we can assume the following conditions for any strategy for a building speed $v < 2$.

$$\begin{aligned} a_{i+1} \geq b_i \quad \text{and} \quad b_{i+1} \geq 2b_i \quad \forall i \geq 1, \\ \text{similarly} \quad c_{i+1} \geq d_i \quad \text{and} \quad d_{i+1} \geq 2d_i \quad \forall i \geq 1. \end{aligned} \tag{3.63}$$

This forces the 0-intervals generated by b_i to be of maximal length of b_i. As discussed in Subsection 3.4.1, this results in a repeating sequence of k-intervals of specific lengths in each direction. In these repeating sequences, each 0-interval of length b_i is later followed by a 3-interval of length b_i before the next 0-interval, always separated by 1-intervals. Thus, the local maxima in the consumption of a single direction always happen at the end of the 3-interval, while the local minima are obtained at the end of the 0-intervals.

The goal is to have the local consumption minima in one direction counter the local maxima in the other. To this purpose, a barrier system is constructed in such a way that the 0-intervals of both directions always appear in an alternating fashion. Following this approach, we will prove the following theorem by constructing a barrier system and showing how its consumption-ratio is bounded.

Theorem 15
In the half-plane model, the fire can be contained with speed $v = \frac{17}{9} = 1.\overline{8}$.

Proof of Theorem 15

Consider the periodic interlacing of time intervals as illustrated in Figure 3.20. There, the ends of the 0-intervals in one direction coincide with the ends of the 3-intervals in the other direction, that is, at t_3 and t_6.

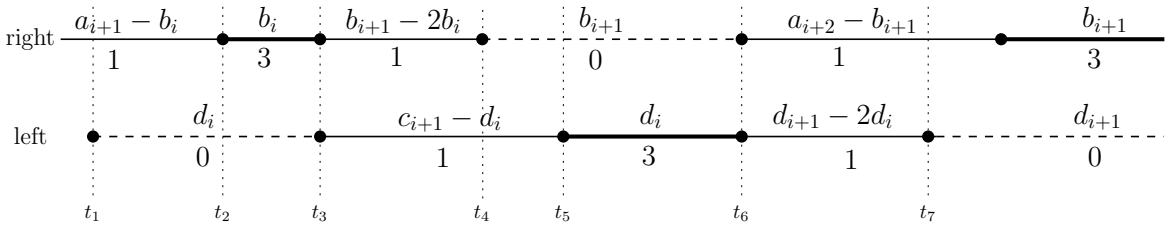


Figure 3.20: The periodic interlacing of time intervals.

The current consumption is always greater than 1, since the 0-intervals do not overlap. Also, the combined consumption-ratio $\mathcal{Q}(t)$ must be smaller than 2 at all times. This also implies that t_3 is no local maximum of $\mathcal{Q}(t)$ as the combined consumption-ratio grows towards 2 between t_3 and t_4. Hence, by setting $d_i > 2b_i$ we make t_1, t_4, t_7 the local maxima and t_2, t_5 the local minima of $\mathcal{Q}(t)$.

Consider the consumption-ratio $\mathcal{Q}(t_1,t_4)$ of the cycle from t_1 to t_4. There are two 1-intervals involved in this cycle in the right direction. The first one, where the fire burns along a_{i+1}, is of length $a_{i+1}-b_i$ and lies partially in this cycle. The second one, where the fire crawls up along b_{i+1}, is of length $b_{i+1}-2b_i$ and lies completely in this cycle.

As the beginning of this cycle is given by the start of the 0-interval on one side and the end is given by the end of the second 1-interval on the other side, the length of this cycle is exactly $d_i+(b_{i+1}-2b_i)$. The total consumption in this cycle is then $d_i+2b_i+2(b_{i+1}-2b_i)$. This means

$$\mathcal{Q}(t_1,t_4)=\frac{\mathcal{C}(t_1,t_4)}{t_4-t_1}=\frac{d_i+2b_i+2(b_{i+1}-2b_i)}{d_i+(b_{i+1}-2b_i)}. \tag{3.64}$$

Now we choose $d_i=\beta\cdot b_i$ and $b_{i+1}=\beta\cdot d_i$ for some $\beta\in\mathbb{R}$ with $\beta>2$. Note that this choice satisfies all our conditions with $b_{i+1}=\beta^2 b_i$ and $d_i>2b_i$. Applying this to Equation 3.64 yields

$$\frac{\mathcal{C}(t_1,t_4)}{t_4-t_1}=\frac{\beta b_i+2b_i+2(\beta^2 b_i-2b_i)}{\beta b_i+\beta^2 b_i-2b_i}=\frac{2\beta^2+\beta-2}{\beta^2+\beta-2}.$$

This expression attains a minimal value of value of $\frac{17}{9}$ for $\beta=4$. Note that by design, $\mathcal{Q}(t_1,t_2)$ and $\mathcal{Q}(t_1,t_3)$ stay below $17/9$, as well. Moreover, the global consumption-ratio is at most $17/9$ at the end of the cycle at t_4, if and only if the global consumption-ratio at the beginning of the cycle at t_1 is at most $17/9$, as

$$\mathcal{Q}(t_4)=\frac{\mathcal{C}(t_1)+\mathcal{C}(t_1,t_4)}{t_4}=\frac{t_1}{t_4}\cdot\frac{\mathcal{C}(t_1)}{t_1}+\frac{t_4-t_1}{t_4}\cdot\frac{\mathcal{C}(t_1,t_4)}{t_4-t_1}\leq\frac{17}{9}.$$

Since the cycles change their roles at t_4 such that the 0-interval occurs in the right direction, the same argument can be used to bound the local consumption-ratio in the following interval and for all subsequent cycles inductively. Note, that by looking at the time interval from t_3 to t_6, we can derive a closed form for c_{i+1}, namely $c_{i+1}=2b_{i+1}-2b_i=(2\beta^2-2)b_i$. Similarly $a_{i+1}=2d_i-2d_{i-1}=(2\beta^2-2)d_{i-1}=\beta c_i$.

It remains to find initial values to get the periodic interlacing started, while maintaining $\mathcal{Q}(t)\leq 17/9$. These provide an induction base, which together with the arguments above, prove $\mathcal{Q}(t)\leq 17/9$ at all t. Suitable values are

$$\begin{array}{lllll} a_1\coloneqq s & b_1\coloneqq 17s & a_2\coloneqq 34s & a_{i+1}\coloneqq 7.5b_i & b_{i+1}\coloneqq 4d_i \\ c_1\coloneqq s & d_1\coloneqq 34s & c_2\coloneqq 238s & c_{i+1}\coloneqq 7.5d_i & d_{i+1}\coloneqq 4b_{i+1}, \end{array}$$

which results in the starting intervals given in Figure 3.21. The local maxima at t_1 and t_4 then have consumption-ratio exactly $17/9$. The interval between t_2 and t_3 is set up equivalent to the one between t_3 and t_6 in Figure 3.20, which means the interlacing construction can be applied to all intervals beyond, which concludes the proof of Theorem 15. Note that all barriers scale with the length of the head-start barrier s.

An example of this construction for $s=1$ is given in Figure 3.22.

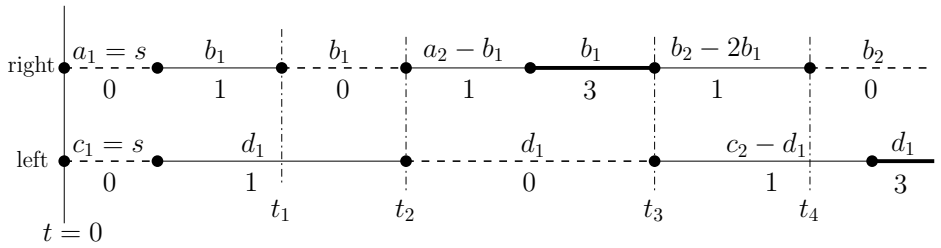


Figure 3.21: Illustration of time intervals at the start. Due to their exponential growth, the sizes of the intervals are not true to scale.

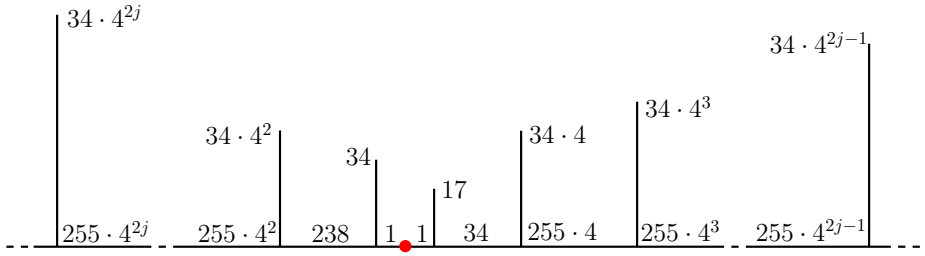


Figure 3.22: Example for the final barrier system for $s = 1$, also not true to scale.

3.5.2 An Improved Strategy

It is possible to improve the barrier system constructed in Subsection 3.5.1 to reduce the upper bound of $v = \frac{17}{9} = 1.\overline{8}$ slightly to obtain the following Theorem:

Theorem 16

In the half-plane model, the fire can be contained with speed $v \geq 1.8772$.

As shown in Figure 3.20, the end of the 3-interval in one direction coincides with the end of the 0-interval in the other direction, which makes t_4 the only local maximum of the interval $[t_1, t_4]$. We introduce a regular shift by a factor of δ, see Figure 3.23. This allows the 3-interval in one direction to lie completely inside the 0-interval of the other direction, as shown in Figure 3.23. Then, there are two local maxima in the equivalent interval $[t_1, t_5]$, namely at t_3 and t_5. We force both maxima to attain the same value to minimize both at the same time.

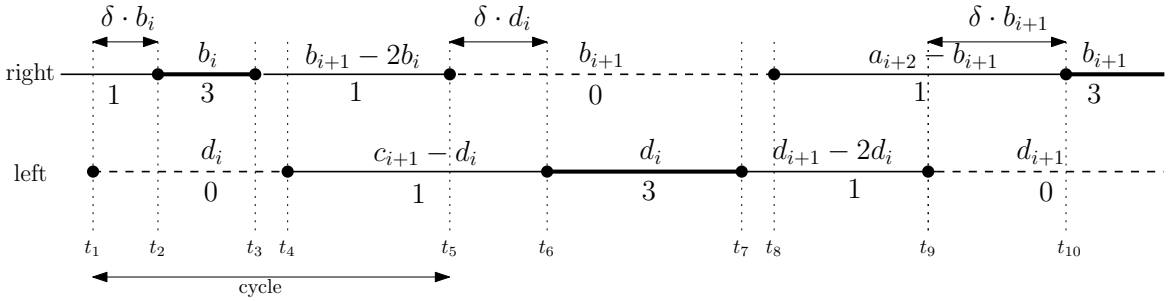


Figure 3.23: A general periodic interlacing of time intervals.

Again, we set $d_i = \beta \cdot b_i$ and $b_{i+1} = \beta \cdot d_i$, for some $\beta \geq 1$ determined below. Then the value of the first local maximum can be expressed as

$$\mathcal{Q}(t_1,t_3) = \frac{\mathcal{C}(t_1,t_3)}{t_3 - t_1} = \frac{1 \cdot (\delta \cdot b_i) + 3 \cdot b_i}{\delta \cdot b_i + b_i} = \frac{\delta + 3}{\delta + 1} = 1 + \frac{2}{\delta + 1}.$$

Considering the cycle from t_1 to t_5 in Figure 3.23, we can conclude that $c_{i+1} = b_{i+1} - b_i + \delta b_i + \delta d_i$. Similarly, we can proceed on the interval from t_5 to t_9 to express a_{i+1} in terms of β, δ and b_i.

Using these identities, we obtain for the second local maximum

$$\begin{aligned}
\mathcal{Q}(t_1,t_5) = \frac{\mathcal{C}(t_1,t_5)}{t_5 - t_1} &= \frac{1 \cdot (\delta \cdot b_i) + 3b_i + 1 \cdot (b_{i+1} - 2b_i) + 1 \cdot (c_{i+1} - d_i - \delta \cdot d_i)}{d_i + (c_{i+1} - d_i) - \delta \cdot d_i} \\
&= \frac{c_{i+1} - \delta \cdot d_i}{c_{i+1} - \delta \cdot d_i} + \frac{b_{i+1} + \delta \cdot b_i + b_i - d_i}{c_{i+1} - \delta \cdot d_i} \\
&= 1 + \frac{b_{i+1} - b_i + \delta \cdot b_i + 2b_i - d_i}{b_{i+1} - b_i + \delta b_i} \\
&= 2 + \frac{2b_i - d_i}{b_{i+1} - b_i + \delta b_i} = 2 + \frac{2 - \beta}{\beta^2 - 1 + \delta}.
\end{aligned}$$

As mentioned above, we set both local maxima to be equal, solve for δ and obtain

$$\delta = \frac{1}{2}\left(\beta - \beta^2 + \sqrt{-12 + 4\beta + 5\beta^2 - 2\beta^3 + \beta^4}\right).$$

Plugging this into either one of the two local maxima and minimizing the resulting function for $\beta \geq 1$, we obtain

$$\beta = \frac{3}{2} + \frac{1}{6}\left(513 - 114\sqrt{6}\right)^{1/3} + \frac{\left(19(9 + 2\sqrt{6})\right)^{1/3}}{2 \cdot 3^{2/3}} \approx 4.06887$$

for the optimal value of β, $\delta \approx 1.2802$ and

$$v = \frac{1}{6}\left(10 - \frac{19^{2/3}}{\sqrt[3]{2(4 + 3\sqrt{6})}} + \frac{\sqrt[3]{19(4 + 3\sqrt{6}))}}{2^{2/3}}\right) \approx 1.8771$$

as the minimum speed.

Note that the optimal value for β satisfies our conditions given in Equation 3.63, so that the barrier system can in fact be realized. Finally, we give suitable values to get the interlacing started:

$$\begin{aligned}
a_1 &\coloneqq s & c_1 &\coloneqq s \\
b_1 &\coloneqq \frac{v}{(4\beta + 2\delta + 1) - v(2\beta + \delta + 1)}\, s & d_1 &\coloneqq 2b_1 \\
a_2 &\coloneqq (\delta + 1) \cdot b_1 & c_2 &\coloneqq (2\beta + 3\delta - 1) \cdot b_1 \\
a_{i+1} &\coloneqq (\delta - 1)d_i + (\beta + \delta)b_{i+1} & c_{i+1} &\coloneqq (\delta - 1)b_i + (\beta + \delta)d_i
\end{aligned}$$

$$b_{i+1} \coloneqq \beta \cdot d_i \qquad\qquad d_{i+1} \coloneqq \beta \cdot b_{i+1}.$$

Note that all barriers scale with the length of the head-start barrier s. This concludes the proof of Theorem 16.

3.6 Additional Considerations and Open Problems

3.6.1 Removing the Starting Buffer

The two strategies constructed in Section 3.5 both utilize the same effect. They use the 0-interval generated by the arbitrarily small constant starting buffer to build the first vertical barrier in one direction. That vertical barrier generates another 0-interval, during which an even bigger vertical barrier is build in the other direction, that generates a bigger 0-intervals and so on. Without the initial buffer, but with the fire starting at the highway - any strategy for $v < 2$ must instantly fail.

In the original setting by Bressan however, the fire does not start with a single point, but with a small circle. This enforces that any enclosing barrier has a minimum distance from the fire origin point and therefore measurable length. This avoids the problem of strategies consisting of immeasurably small barriers around an origin point, which can be imagined as stepping on the fire before it actually breaks out.

An intuitive way to adapt this would be to have the fire start at some distance s to the horizontal boundary of the half-plane. This way, we also get to start with a 0-interval of length s before consumption begins in both directions. The question, is whether the strategies working in the half-plane model with the head-start buffer can be adapted to this variant. Due to the similarities, we can use all notations regarding barrier lengths, consumption, etc. as for the half-plane model. As soon as the fire has passed some vertical barrier b_i with $b_i > s$, the further consumption in that direction is identical in both settings, so the answer to this question is only dependent on the handling of the starting situation. Interestingly, these two settings are more distinct than their obvious similarities suggest, as the following theorem shows.

Theorem 17

Assume a fire starts at distance s from the boundary of the half-plane and spreads with unit speed according to the L_1-metric. Then there exists no barrier system containing the fire in the upper half-plane with speed $v < 2$ consisting only of a horizontal barrier along the boundary of the half-plane and vertical barriers attached to it.

Proof. The first crucial observation is that for any sensible strategy, the first vertical barrier in both directions must have length at least s, so $b_1 > s$ and $d_1 > s$. The argument is the same as for strictly increasing length of vertical barriers, as a barrier shorter than the initial distance s is just passed over by the fire without delaying it.

Assume that there exists a strategy that contains the fire for a building speed $v < 2$, We will consider the times when the fire reaches the end of a horizontal piece of barrier in either direction, as these mark the end of 2- or 3-interval in that direction and hence, local maxima in consumption. Independent of the actual lengths and positions of the vertical barriers, we will show that at latest at the third such time, the consumption-ratio exceeds 2.

W. l. o. g., assume $a_1 \leq c_1$, so the first vertical barrier encountered by the fire lies in the right direction, as illustrated in figure Figure 3.24. Let t_1 be the moment in time, when the fire reaches the end of the horizontal barrier a_i and, simultaneously, the bottom end of b_1, i. e., $t_1 = a_1 + s$. Then $\mathscr{C}^r(t_1) = a_1 + \min\{b_1, 2s\}$ and $\mathscr{C}^l(t_1) \geq t_1 - s = a_1$ and hence, $\mathscr{C}(t_1) \geq 2a_1 + \min\{b_1, 2s\}$. This means $\mathscr{C}(t_1) < 2t_1$ only if $b_1 < 2s$.

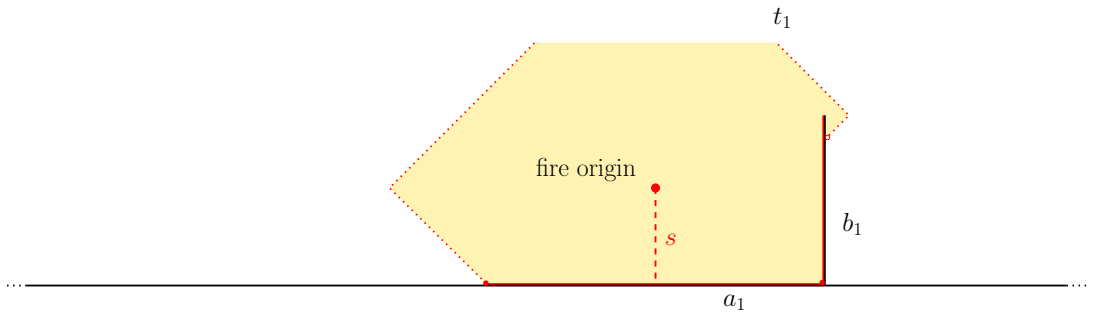


Figure 3.24: The situation at time t_1, when the fire reaches the end of a_1.

Let t_2 be the first moment in time after t_1, when the fire reaches the end of a horizontal piece of barrier. We have to consider two cases. Either this happens in the same direction as the first time, i.e. the right direction (Case 1) - or in the opposite direction (Case 2).

Case 1: This case is illustrated in Figure 3.25. Then $t_2 = a_1 + a_2 + 2b_1 - s$, $\mathscr{C}^r(t_2) = a_1 + a_2 + b_1 + \min\{b_2, 2b_1\}$ and $\mathscr{C}^l(t_2) \geq t_2 - s = a_1 + a_2 + 2b_1 - 2s$. In summary, $\mathscr{C}(t_2) \geq 2a_1 + 2a_2 + 3b_1 - 2s + \min\{b_2, 2b_1\} = 2t_2 + \min\{b_2 - b_1, b_1\} > 2t_2$, which is a contradiction to the assumption, that this strategy contains the fire for $v < 2$.

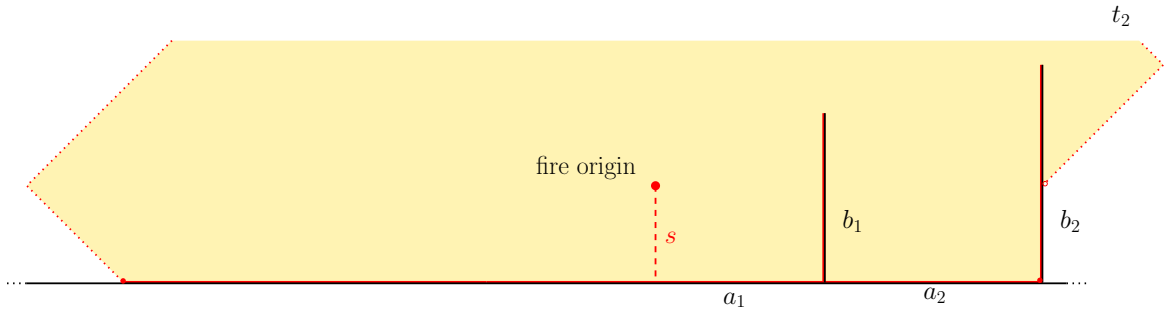


Figure 3.25: The situation at t_2 in Case 1, when the fire reaches the end of a_2.

Case 2: This case is illustrated in Figure 3.26 By definition, $t_2 = c_1 + s$ and, by $t_2 > t_1$, $t_2 = a_1 + b_1 + (b_1 - s) + x$ for some $x \geq 2s - 2b_1$. Observe that x can be negative, if the fire has not yet reached the next horizontal barrier piece a_2. Either way, $\mathscr{C}^r(t_2) \geq a_1 + b_1 + x = t_2 - (b_1 - s)$. In the left direction $\mathscr{C}^l(t_2) = c_1 + \min\{d_1, 2s\} = t_2 + \min(d_1 - s, s)$. In summary, $\mathscr{C}(t_2) = 2t_2 + \min\{d_1 - s, s\} - (b_1 - s)$. Therefore $\mathscr{C}(t_2) < 2t_2$ only if $min\{d_1 - s, s\} < b_1 - s < s \Rightarrow d_1 < 2s$.

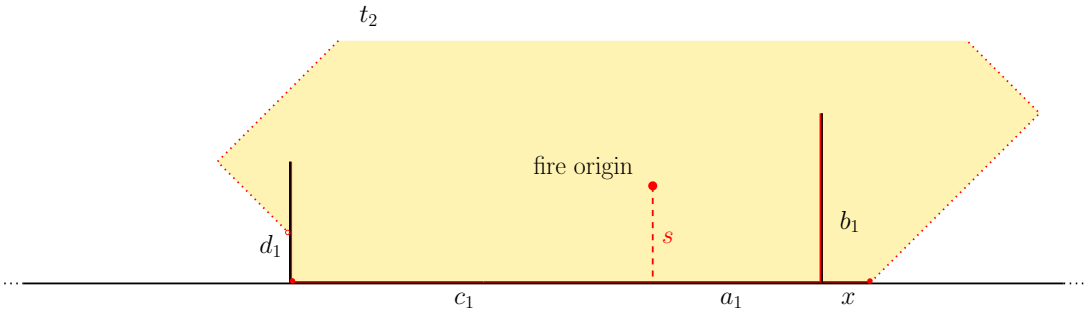


Figure 3.26: The situation at t_2 in Case 2, when the fire reaches the end of c_1.

We have established that both $b_1 < 2s$ and $d_1 < 2s$. Now let t_3 be the third time, when the fire reaches the end of a horizontal piece of barrier. The argument works identical for either direction, so we assume w. l. o. g. this happens in the right direction. It is illustrated in Figure 3.27.

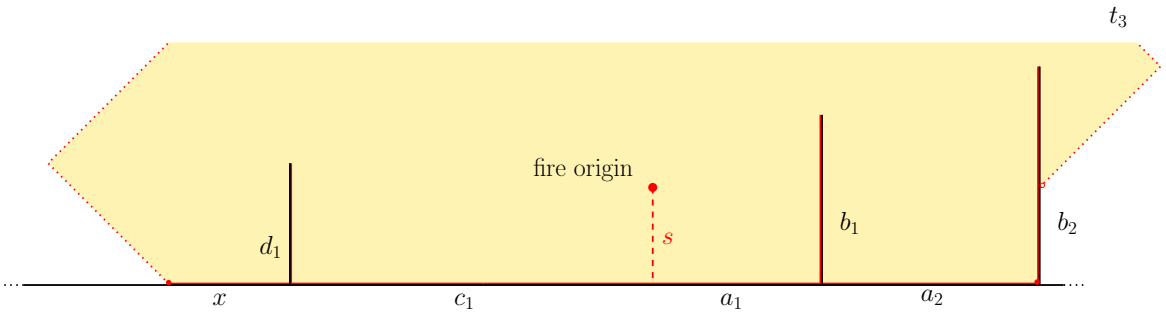


Figure 3.27: The situation at t_3, when the fire reaches the end of a_2.

By definition, $t_3 = a_1 + a_2 + 2b_1 - s$ and, by $t_3 > t_2$, $t_3 = c_1 + d_1 + (d_1 - s) + x$ for some $x \geq 2s - 2d_1$. Again, x can be negative if the fire has not yet reached the next horizontal barrier piece c_2. Either way, $\mathscr{C}^l(t_3) \geq C_1 + d_1 + x = t_3 - (d_1 - s)$. In the right direction, $\mathscr{C}^r(t_3) = a_1 + a_2 + b_1 + \min\{b_2, 2b_1\} = t_3 + s + \min\{b_2 - b_1, b_1\}$. In summary, $\mathscr{C}(t_3) = 2t_3 + s + \min\{b_2 - b_1, b_1\} - (d_1 - s) > 2t_3$ as $s > d_1 - s$ and $b_2 > b_1$, which is a contradiction to the assumption that this strategy contains the fire for $v < 2$. □

3.6.2 Closing the Gap

There remains a gap between known lower bound of $v > 1.\overline{6}$ necessary to contain the fire in the half-plane model (see Subsection 3.4.4) and the speed $v > 1.8772$ necessary for the best known strategy (see Subsection 3.5.2). The proof of the lower bound utilizes several bounds on the length of

specific barriers and intervals to derive contradictions for speeds $v \leq 1.\overline{6}$. However, the inequalities representing those bounds still include some slack, so it is intuitive to assume that the true lower bound on v is stronger than what we could show here. Besides tightening the inequalities further, another way to close the gap from below would be to show that the best known strategy is actually optimal.

The strategy presented in Subsection 3.5.2, is optimizing a strategy constructed based on several structural assumption. First, there is the general structural assumption of $b_{i+1} \geq 2b_i$ and $a_{i+1} \geq b_i$. Second is the assumption of recursive regularity, i. e. that after some constant number of initializing barriers, $b_{i+1} = \beta d_i = \beta^2 b_i$. Third is the structure of the regular interlacing, i. e. that the 3-interval in the one direction lies completely in the 0-interval in the other direction.

Among all strategies fulfilling these assumptions, the strategy presented in Subsection 3.5.2 is optimal. Hence showing that there is an optimal strategy fulfilling these assumptions would close the gap.

We already showed in Subsection 3.4.1, that we can assume an optimal barrier to fulfil $b_{i+1} \geq 2b_i$ and $a_{i+1} \geq b_i$. The core of this open problem are the assumptions on the regularity of the barrier system, mainly the second assumption.

Given the second assumption, one can rule out all other possible ways of interlacing. The proof of Lemma 12 can be adapted to show that the directional consumption-ratio exceeds 1 at any time during the 1-intervals of length $b_{i+1} - 2b_i$ and $d_{i+1} - 2d_i$ respectively. Therefore, these two intervals cannot overlap in the interlacing, which limits the possible interlacings to a small number of cases. Optimizing each case similar to the one in Subsection 3.5.2) yields no better strategy.

But as long as one cannot show that regular recursive growth is optimal in general, the gap remains open.

3.6.3 Higher Dimensions

Considering a problem in higher dimension is a natural extension for any geometric problem. For the canonical geometric firefighting model, moving from $\mathbb{R}^2$ to $\mathbb{R}^3$ naturally would have the fire spread in all directions according to the unit sphere. However, 1-dimensional barrier curves can not contain such a 3-dimensional fire. Therefore, barriers to contain the fire must then be two-dimensional surfaces in the Euclidean space, bounded in area by a building speed v. So far, no known progress has been made towards any bounds in this setting.

The half-plane model however, lends itself to a different way of extending the dimension. During the analysis it might be noticed, that the spread of the fire in either direction is completely independent on any barrier in the other direction. So instead of regarding it as two directions of one half-plane, one can also regard both directions as as two separate quarter planes. The fire starts in the corner of each quarter plane and spreads through both in parallel. So instead of considering containing the fire above one horizontal line, one could just as well consider it as containing the fire above two horizontal rays in parallel.

From there on, it is easy to extend the half-plane model with a parameter k defining the number of directions (or quarter planes) the fire spreads along in parallel. As for $k = 2$ as studied until now, we assume a head-start barrier of small constant length already built along the beginning of each horizontal ray. It is trivial to see, that speed k is enough to just build horizontal barriers along each ray. In fact, by *Lemma* 12, this is optimal for $k = 1$. But our strategies for the half-plane model show, that we can do better for $k = 2$. For even $k \geq 2$, one can adapt the strategy given in Subsection 3.5.2 to k pairs of direction to contain the fire in all directions for a speed of $\frac{k}{2}1.8772$. For odd $k \geq 1$, the same approach in combination with just building a horizontal barrier along the remaining unpaired direction yields a strategy for a speed of $\frac{k-1}{2}1.8772 + 1$.

The lower bound proof however, can not be directly extended in a similar manner, as it remains open whether there is a more clever way to interweave the consumption intervals of all directions to create a strategy for an even lower speed.

3.6.4 Euclidean Metric

Another natural extension of the half-plane model would be to have the fire spread according to the euclidean metric instead of the L_1-metric. This would bring it closer to Bressan's original geometric firefighting problem.

A fire spreading uniformly in all directions according to the L_2-metric is strictly stronger than one spreading according to the L_1-metric, as the L_1 distance between two points is always greater or equal to the L_2 distance. Therefore, all lower bounds for barrier systems consisting of a horizontal barrier with vertical barriers attached to it as obtained in Section 3.4, hold for the L_2-setting as well. However, it is less intuitive in the L_2 setting to restrict the barrier systems to vertical delaying barriers only.

It is also clear that the strategies used for the constructive upper bound in Section 3.5 require a higher speed in the L_2-setting as the construction is tight for some times t. However, for a given barrier system it is not easy to see what the required speed is, even if it only consists of the horizontal enclosing barrier and vertical delaying barriers. The exception of course is the trivial barrier system only consisting of the horizontal barrier, which requires speed $v = 2$.

For analysing more complex barrier systems there are two main difficulties, when trying to apply the consumption-ratio approach. It still holds that $sup_t \mathscr{Q}(t) < vt$ is necessary and sufficient for a barrier system to contain the fire in the upper half-plane. However, as a crucial difference, there are no longer intervals of constant current consumption as heavily utilized in all proofs for the half-plane model (except when the consumption is 0).

Consider for example, a vertical barrier of length $b_1$has been built right at the end of the head-start barrier of length s. Then its lower end is reached at time s and its upper end at time $\sqrt{b_1^2 + s^2}$. At any point in between, it holds by the Pythagorean theorem $\mathscr{C}(t) = \sqrt{t^2 - s^2}$. The speed with which the fire burns along b_1, i. e. the current consumption, at such a time t is the derivative of the consumption, hence $\mathscr{Q}(t) = \frac{x}{x^2 - s^2}$, which is far from constant. But still, the consumption during that interval is governed by the derivative of a single, simple function.

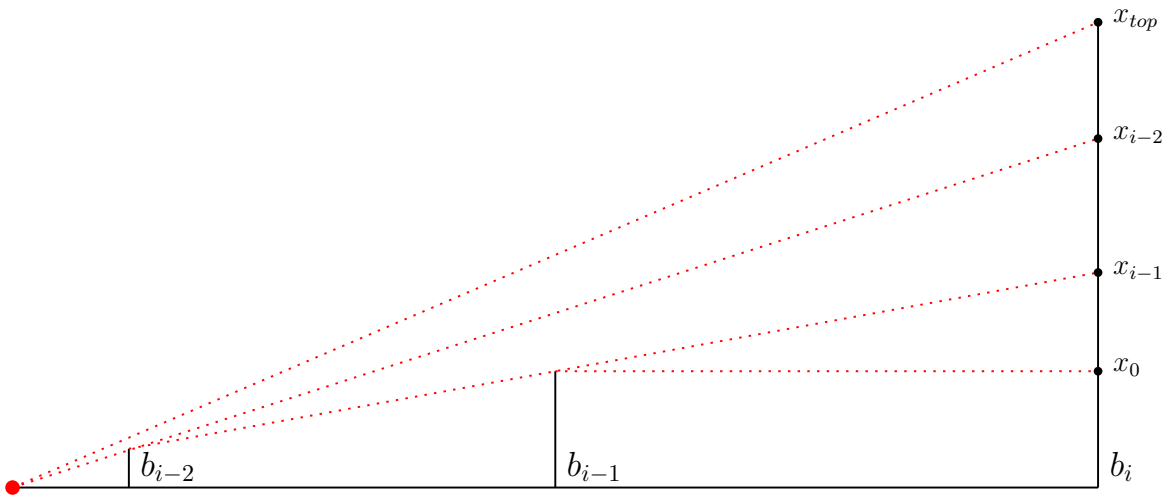


Figure 3.28: An example of a barrier b_i which consumption speed is piecewise dependent on all previous barriers. In between x_0 and x_{i-1}, the previous barrier b_{i-1} delays the fire. In between x_{i-1} and x_{i-2}, the previous barrier b_{i-2} delays the fire. From x_{i-2} up to the top, no previous vertical barrier delays the fire at all.

This is no longer true for later barriers, whose consumption can be governed by a complicated multipart function of all previous barriers b_j and a_j with $j < i$. This even holds, if we still assume $b_i > 2b_{i-1}$. As in the half-plane model, this guarantees, that all barriers a_j and b_j with $j < i$ have been completely consumed when the fire reaches the top of b_i. However, consider a barrier b_i as illustrated in Figure 3.28.

In the half-plane model, a point at height $x > b_{i-1}$ on the barrier b_i is always reached at time $A_i + x$ and a point with $x < b_i$ at time $A_i + 2b_{i-1} - x$. When the fire spreads according to the L_2-metric, however, barriers b_j with $j < i-1$ can also affect when a point on the barrier b_i is reached. If there were no barriers b_1 to b_{i-1}, the point at height $x > b_i$ on the barrier is reached exactly at time $t_x = \sqrt{A_i^2 + x^2}$. However, if the straight line from the fire origin to x intersects some of the previous barriers, these delay the fire and affect the consumption of x. The shortest path from the fire to x is then not a straight line segment, but a series of line segments connecting the fire origin with x via several of the top ends of the intersected barriers. But as the fire climbs higher along b_i, the straight line from the fire origin to x will intersect less and less previous barriers, resulting in the complexity of the shortest path to x decreasing. As in the example in Figure 3.28, the shortest path to a point x might be influenced by each previous barrier, which means that the consumption of b_j above a height of b_{i-1} can be governed piecewise by i different functions.

Finally, all shortest paths from the fire to some point x on the barrier system and thus, the time at which x is reached, will be expressed in sums of square roots, which are just generally harder to work with.

All this makes it very hard to pinpoint local maxima in consumption and find a recursive strategy similar to Subsection 3.5.2 which contains the fire for a speed $v < 2$. It remains open to find a barrier

system better than the trivial one for $v = 2$ and shrink the gap between the lower bound of 1.66 and the upper bound of 2 for the speed v necessary to contain a fire spreading according to the L_2-metric in the upper half plane with a barrier system consisting of a horizontal barrier and vertical barriers attached to it. Analysis of differently structured barrier systems is also open for this setting.

Chapter 4

The Minimum Enclosing Ball Problem

In this chapter we present a (1+ε)-approximation algorithm for the Minimum Enclosing Ball problem.

Minimum Enclosing Ball (MEB)
Input: *A set of points $P \subset \mathbb{R}^d$.*
Goal: *Find a ball containing P with minimum radius. This is equivalent to finding the center $c \in \mathbb{R}^d$ minimizing $\max_{p \in P} d(c,p)$, where $d(c,p)$ denotes the Euclidean distance between c and p.*

Throughout this chapter, we will use the following notations: Let $B(c,r)$ denote a ball of radius r centered at c and let $r(B)$ and $c(B)$, denote the radius and center of a ball B, respectively. We denote by pq the straight line segment between two points p and q and by $|pq|$ the length of pq. Finally, we denote the boundary of a closed set A by ∂A.

We will refer to the optimal solution of MEB for a set P as $B^* = B(c^*, r^*)$. A center c is a $(1+\varepsilon)$-approximation, if $P \subset B(c,(1+\varepsilon)r^*)$, i. e., if $\max_{p \in P} d(c,p) \leq (1+\varepsilon)r^*$. By the triangle inequality, an arbitrary point c has distance at most $|cc^*| + r^*$ to any point in P. Therefore, c is a $(1+\varepsilon)$-approximation of B^* if and only if $|cc^*| \leq \varepsilon r^*$, i. e., $c \in B(c^*, \varepsilon r^*)$.

The general approach used both by our algorithm as well as some previously known algorithms like the core-set based approaches and the naive gradient-descent algorithm [7–9, 61] is sketched in Algorithm 3. Essentially, starting with an arbitrary point c in P as an initial candidate, one repeatedly uses the farthest point f_c from c in P to improve c, as the distance between f_c and c dictates the approximation factor of c. The core-set based algorithms do so by adding f_c to the core-set used to construct the candidate center; while the gradient-descent approach moves the center c on the straight line towards f_c, with decreasing step size each round.

Our algorithm follows the gradient-descent approach, so we pick a new center c_i in round i somewhere on the straight line segment between c_{i-1} and $f_{c_{i-1}}$. We start with the proof of a geometric property of line segments in balls, that the exact construction of the next step is based on. We will subsequently present the construction of the new center in each round. To analyse the run-time of our algorithm we determine an upper bound on the number of rounds until our algorithm arrives at

Algorithm 3: General MEB Approach

Input : Set of points $P \subset \mathbb{R}^d$.
Output : A center c such that $B(c,(1+\varepsilon)r^*)$ covers P

```
1 pick an arbitrary point from P as center c
2 while c is a not a (1+ε)-approximation do
3 |   find farthest point f_c from c in P
4 |   construct a better center c from c and f_c
5 return c
```

a $(1+\varepsilon)$-approximation. Finally we discuss how to extend these results to the Euclidean k-center problem.

4.1 A Structural Property

The Structural Property central to our algorithm is a generalization of an idea used for a Euclidean 2-center streaming algorithm by Kim and Ahn [47]. They observe that the midpoint of any line segment of length at least 1.2 placed completely inside a unit Ball B_0 has distance at most 0.8 to the center of B_0. We generalize this observation in several ways.

Theorem 18
For $d \geq 2$, let $B = B(c,r)$ and $B' = B(c,r')$ be two d-dimensional balls with the same center c, such that $r > r'$. Let pp' be a line segment in B with $|pp'| \geq r$, such that $p \in \partial B$ and $p' \in \partial B'$ are points on the boundaries of B and B'. Let B'' be the unique d-dimensional ball centered around c that is tangential to pp'. Let m be the tangential point of B'' and pp'. Let $q \in B$ and $q' \in B'$ be points in B and B', such that $|qq'| > |pp'|$.

Then any point m^* on qq' with $|qm^*| \geq |pm|$ and $|q'm^*| \geq |p'm|$ lies inside B''.

Proof. For $d = 2$ we first show that qq' intersects B'' at all. Note that all pairs $p \in \partial B$ and $p' \in \partial B'$ with the same distance $d(p,p')$ are tangential to the same Ball B''. Hence, we can assume without loss of generality, that p', c and q' are collinear, and also $q' \in cp'$. Because of $|qq'| > |pp'|$, point q must lie in $B \setminus B(p',|pp'|)$, which means qq' intersects B'' as illustrated in Figure 4.1.

Let m^* be an arbitrary point on qq' with $|qm^*| \geq |pm|$ and $|q'm^*| \geq |p'm|$. Assume m^* is not contained in B''. As qq' intersects B'', $qm^* \cap B'' = \emptyset$ and $q'm^* \cap B'' \neq \emptyset$ or vice-versa.

Let us consider $qm^* \cap B'' = \emptyset$ and $q'm^* \cap B'' \neq \emptyset$ first. In this case $q \in B \setminus B''$.

Let q_t be the point on the boundary of B'' such that qq_t is tangential to B'' and the segment qq' lies in the wedge spanned by q_t and c with its tip in q. As $qm^* \cap B'' = \emptyset$, m^* lies within the triangle qq_tc and $|qq_t| > |qm^*|$. But by construction of B'', $|pm| \geq |qq_t|$, which contradicts $|pm| \leq |qm^*|$. This is illustrated in Figure 4.2 (assuming without loss of generality that p is collinear with qq' by similar arguments as above). The other case ($q'm^* \cap B'' = \emptyset$ and $qm^* \cap B'' \neq \emptyset$) can be shown equivalently by switching q and q' and replacing p and B with p' an B', respectively.

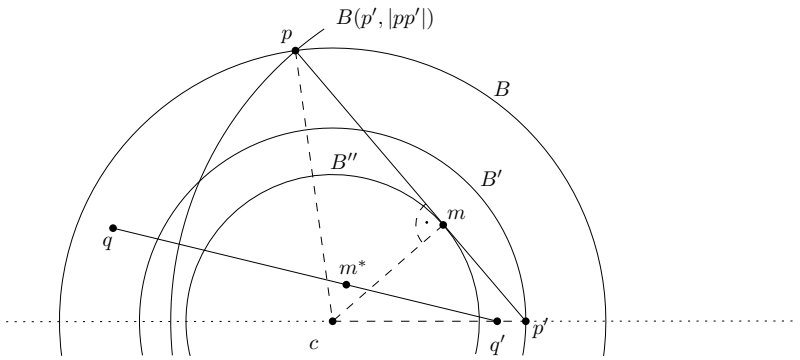


Figure 4.1: Construction of m and B''. Segment qq' must intersect B'' if $|qq'| \geq |pp'|$.

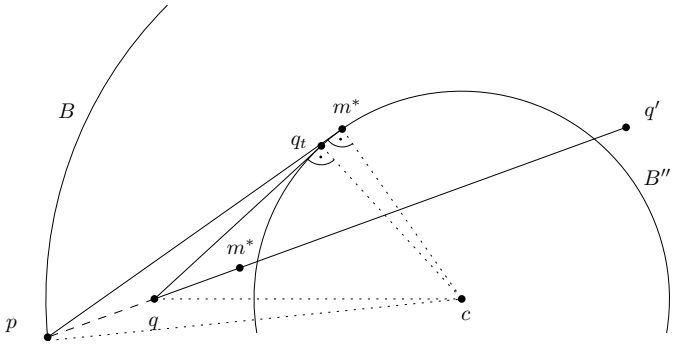


Figure 4.2: Assume $qm^* \cap B'' = \emptyset$. Then $|qm^*| < |qq_t| \leq |pm|$. This contradicts $|qm^*| \geq |pm|$.

For $d > 2$, the same arguments as for $d = 2$ hold in the 2-dimensional plane passing through c and qq'. □

Our algorithm will make use of a special case of this core Theorem: Let the point m' be at the same relative position on qq' as m is on pp', i. e. $\frac{|m'q|}{|qq'|} = \frac{|mp|}{|pp'|}$ and $\frac{|m'q'|}{|qq'|} = \frac{|mp'|}{|pp'|}$. Then m' fulfils the conditions $|qm'| \geq |pm|$ and $|q'm'| \geq |p'm|$ from Theorem 18. We can calculate the radius of B'' as the height of the triangle pcp' using Herons formula, and calculate the relative length $\frac{|mp|}{|pp'|}$ by applying the Pythagorean Theorem. Together, this yields the following corollary:

Corollary 1
For $d \geq 2$, let $B = B(c,r)$ and $B' = B(c,r')$ be two d-dimensional balls with the same center c, such that $r' = \delta r$ for some $0 < \varepsilon \leq \delta \leq 1$. Let $q \in B$ and $q' \in B'$ with $|qq'| \geq (1+\varepsilon)r$.

Then the ball $B'' = B(c,r'')$ with

$$r'' = r\frac{\sqrt{((1+\varepsilon)+1+\delta)\,((1+\varepsilon)+1-\delta)\,((1+\varepsilon)-1+\delta)\,(-(1+\varepsilon)+1+\delta)}}{2\,(1+\varepsilon)}$$
$$= r\frac{1}{2\,(1+\varepsilon)}\sqrt{\left((2+\varepsilon)^2-\delta^2\right)(\delta^2-\varepsilon^2)}$$

is tangential to any line segment pp' of length $|pp'| = (1+\varepsilon)r$ with $p \in \partial B$ and $p' \in \partial B'$ at a point m, and contains the point

$$\begin{aligned} m' &= q + (q'-q)\frac{|mp|}{|pp'|} = q + (q'-q)\frac{\sqrt{r^2 - r''^2}}{(1+\varepsilon)\,r} \\ &= q + (q'-q)\frac{1}{1+\varepsilon}\sqrt{1 - \left(\frac{\sqrt{\left((2+\varepsilon)^2 - \delta^2\right)(\delta^2 - \varepsilon^2)}}{2(1+\varepsilon)}\right)^2} \\ &= q + (q'-q)\frac{1}{2(1+\varepsilon)^2}\sqrt{2^2(1+\varepsilon)^2 - \left((2+\varepsilon)^2 - \delta^2\right)(\delta^2 - \varepsilon^2)} \end{aligned}$$

4.2 An Efficient Gradient-Descent Algorithm

Our algorithm uses a sequence of points $p_0, p_1, p_2, \ldots$ to calculate a sequence of centers $c_0, c_1, c_2, \ldots$ until we arrive at a $(1+\varepsilon)$-approximation. As sketched in Algorithm 3, p_0 is chosen as an arbitrary point in P, while all subsequent p_i are the farthest point in P from the previous center c_{i-1}.
Let

$$c_i := \begin{cases} p_0, & \text{if } i = 0. \\ m'(p_i, c_{i-1}) := p_i + (c_{i-1} - p_i)\frac{1}{2(1+\varepsilon)^2}\sqrt{2^2(1+\varepsilon)^2 - \left((2+\varepsilon)^2 - \delta_i^2\right)\left(\delta_i^2 - \varepsilon^2\right)}, & \text{otherwise.} \end{cases} \tag{4.1}$$

with

$$\delta_i := \begin{cases} 1, & \text{if } i = 0. \\ \frac{1}{2(1+\varepsilon)}\sqrt{\left((2+\varepsilon)^2 - \delta_{i-1}^2\right)\left(\delta_{i-1}^2 - \varepsilon^2\right)}, & \text{otherwise.} \end{cases} \tag{4.2}$$

Lemma 18
c_i lies in the Ball $B(c^*, \delta_i r^*)$. Thus, it is a $(1+\delta_i)$-approximation.

Proof by Induction. For $i = 0$, $c_0 = p_0$ is a point from P and therefore lies in the optimal Ball $B^* = B(c^*, r^*) = B(c^*, \delta_0 r^*)$. Now assume c_i lies in the Ball $B(c^*, \delta_i r^*)$ for an arbitrary, but fixed i, but is not a $(1+\varepsilon)$-approximation. As $p_{i+1} = f_{c_i}$ and c_i is not a $(1+\varepsilon)$-approximation, $|c_i p_{i+1}| > (1+\varepsilon)r^*$. Also, $p \in B(c^*, r^*)$ as B^* contains all of P. Then $c_{i+1} \in B(c^*, \delta_i r^*)$ follows directly from Corollary 1 with $B = B^*$, $B' = B(c^*, \delta_i r^*)$, $p_i = q$ and $c_i = q'$. □

Note, that it is not obvious how to test whether a center c_i is a $(1+\varepsilon)$-approximation. In fact, the arbitrary point p_i chosen in the very first round could already be a $(1+\varepsilon)$-approximation. While the minimum radius of a ball around c_i that contains P is given by $d(c_i, f_{c_i})$, its relative quality in comparison to r^* can not be determined without knowing r^*. However, if we can bound the number of rounds k it takes at most until our algorithm arrives at a $(1+\varepsilon)$-approximation, running our algorithm

for k rounds and taking the center with the smallest necessary radius among all centers $c_1, c_2, \ldots, c_k$ returns a $(1+\varepsilon)$-approximation.

It remains to find a sufficient bound on the number of rounds. Obviously, $\delta_i < \delta_{i-1}$. In the worst case, c_i lies exactly at the boundary of $B(c^*, \delta_i r^*)$ in each round, such that c_i is a $(1+\varepsilon)$-approximation if and only if $\delta_i < \varepsilon$. Thus, instead of choosing $p_i = f_{c_{i-1}}$, we examine a slightly different sequence of points, that creates this worst case. The length of such a worst-case sequence then gives an upper bound on the number of rounds necessary in our algorithm.

Let $p_1, p_2, \ldots p_k$ be a sequence of points, such that $p_i \in \partial B^*$ and $|p_i c_{i-1}| = (1+\varepsilon)r^*$, where c_i is defined as above. Then the construction in Corollary 1 is tight, as $p_i c_{i-1}$ is exactly tangential to $B(c^*, \delta_i)$, and thus $c_i \in \partial B(c^*, \delta_i)$ is exactly the tangential point. Then, for $i \geq 1$, we consider the line segments $a_i = c_{i-1}c_i$ with $|a_i| = \alpha_i(1+\varepsilon)r^*$ and $b_i = c_i p_i$ with $|b_i| = \beta_i(1+\varepsilon)r^*$ that together form $c_{i-1}p_i$ as illustrated in Figure 4.3.

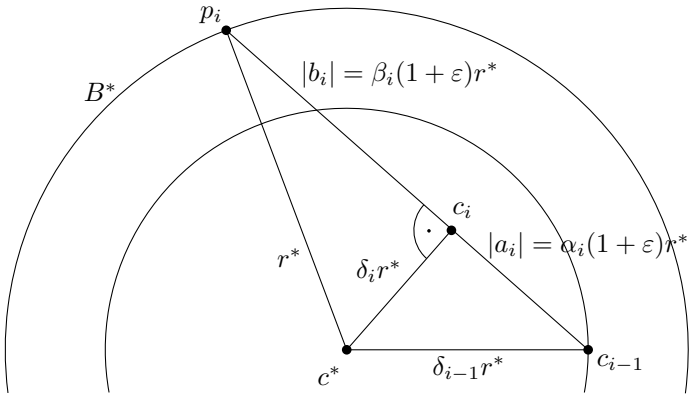


Figure 4.3: Construction of c_i based on c_{i-1} and p_i.

As c_i converges towards c^* with increasing i, β_i increases and α_i decreases. However, β_i can be at most $\frac{1}{(1+\varepsilon)}$ by construction as b_i forms a right-angled triangle with $c^* p_i$ as the hypotenuse, so $\beta_i(1+\varepsilon)r^* = |b_i| < |c^* p_i| = r^*$. We will now show a lower bound on β_i and prove that it exceeds $\frac{1}{(1+\varepsilon)}$ for $i \geq \frac{2}{\varepsilon} - 1$. Thus, there exists no point p_i with $|m_{i-1}p_i| = (1+\varepsilon)r^*$ and $p_i \in \partial B^*$ for $i \geq \frac{2}{\varepsilon} - 1$. This means, the intersection of ∂B^* and $\partial B(m_{i-1}, (1+\varepsilon)r^*)$ is empty, and therefore, $B(c_{i-1}, (1+\varepsilon)r^*)$ covers B^*. That means, our sequence $p_1, p_2, \ldots p_k$ can contain at most $\lfloor \frac{2}{\varepsilon} \rfloor$ many points without any of the corresponding c_i being a $(1+\varepsilon)$-approximation.

Lemma 19
β_i exceeds $\frac{1}{(1+\varepsilon)}$ for $i \geq \lfloor \frac{2}{\varepsilon} \rfloor > \frac{2}{\varepsilon} - 1$.

Proof. By our construction

$$|b_i| = (1+\varepsilon)r^* - |a_i| \qquad \Rightarrow \qquad \beta_i = 1 - \alpha_i. \tag{4.3}$$

In addition, by the construction of c_i as illustrated in Figure 4.3 and the Pythagorean theorem, it holds

$$\begin{aligned}\beta_i^2(1+\varepsilon)^2 &= 1^2 - \delta_i^2 \\ &= 1 - (\delta_{i-1}^2 - \alpha_i^2(1+\varepsilon)^2) \\ &= 1 - ((1-\beta_{i-1}^2(1+\varepsilon)^2) - \alpha_i^2(1+\varepsilon)^2) \\ &= \alpha_i^2(1+\varepsilon)^2 + \beta_{i-1}^2(1+\varepsilon)^2 \\ \Rightarrow \qquad \beta_i^2 &= \alpha_i^2 + \beta_{i-1}^2. \end{aligned} \tag{4.4}$$

Combining Equation 4.3 and Equation 4.4

$$\begin{aligned} 1-\alpha_i &= \sqrt{\beta_{i-1}^2 + \alpha_i^2} \\ \Rightarrow \qquad \alpha_i &= \frac{1-\beta_{i-1}^2}{2}. \end{aligned} \tag{4.5}$$

Applying Equation 4.3 again we obtain the recurrence

$$\beta_i = 1 - \frac{1-\beta_{i-1}^2}{2} = \frac{1+\beta_{i-1}^2}{2}. \tag{4.6}$$

If we substitute $\gamma_i = \frac{1}{1-\beta_i} \Leftrightarrow \beta_i = \frac{\gamma_i - 1}{\gamma_i}$ in Equation 4.6, we get the recurrence

$$\gamma_i = \frac{\gamma_{i-1}}{1 - \frac{1}{(2\gamma_{i-1})}} = \gamma_{i-1}(1 + \frac{1}{2\gamma_{i-1}} + \frac{1}{4\gamma_{i-1}^2} + \cdots) \geq \gamma_{i-1} + \frac{1}{2}. \tag{4.7}$$

Now, p_0 and p_1 both lie on ∂B^*, and thus, $\beta_1 = \frac{1}{2}$ and consequently $\gamma_1 = 2$.
By induction, $\gamma_i \geq \frac{(3+i)}{2}$ and hence

$$\beta_i = \frac{\gamma_i - 1}{\gamma_i} = 1 - \frac{1}{\gamma_i} \geq 1 - \frac{2}{3+i} = \frac{1+i}{3+i} \tag{4.8}$$

Thus, it suffices to have $i \geq \frac{2}{\varepsilon} - 1$ to have

$$\beta_i \geq \frac{1+i}{3+i} \geq \frac{\frac{2}{\varepsilon}}{2+\frac{2}{\varepsilon}} = \frac{\frac{2}{\varepsilon}}{\frac{2\varepsilon}{\varepsilon} + \frac{2}{\varepsilon}} = \frac{1}{1+\varepsilon}. \tag{4.9}$$

□

The upper bound for the number of rounds together with the specific construction of the next center in each step yields the correctness of Algorithm GRADIENTMEB. It remains to analyse its runtime: Finding $p_i = f_{c_{i-1}}$ each round takes time in $O(nd)$ and the computation of m_i takes time in $O(d)$, as its a linear combination of two d-dimensional points. As the algorithm runs for $\lfloor \frac{2}{\varepsilon} \rfloor$ rounds, this concludes the proof of the following Theorem:

Theorem 19

Algorithm GRADIENTMEB *computes a* $(1+\varepsilon)$*-approximation of the Minimum Enclosing Ball in time* $O(\frac{nd}{\varepsilon})$.

Algorithm 4: GRADIENTMEB

Input : Set of points $P \subset \mathbb{R}^d$.
Output : A center c such that $B(c,(1+\varepsilon)r^*)$ covers P

```
1  p_1 ← arbitrary point from P
2  c_1 ← p_1
3  bestRadius ← ∞
4  for i = 1 to ⌊2/ε⌋ do
5  |   p_{i+1} ← farthest point from c_i in P
6  |   if |c_i p_{i+1}| < bestRadius then
7  |   |   bestCenter ← c_{i-1}
8  |   |   bestRadius ← |c_{i-1} p_i|
9  |   c_{i+1} ← m'(p_{i+1}, c_i)
10 return bestCenter
```

4.3 Extension to the k-center Problem for $k > 2$

The algorithm GRADIENTMEB can be extended to solve the Euclidean k-center problem for $k > 2$ by a similar approach as Bădoiu et al. [7] employ for their core-set algorithm.

Euclidean k-center

Input: *A set of points* $P \subset \mathbb{R}^d$ *and a value* $k \in \mathbb{N}$.

Goal: *Find* k *centers* $c_1, \dots c_k \in \mathbb{R}^d$ *minimizing* $\max_{p \in P} \min_{i=1}^{k} d(c_i, p)$*, where* $d(c_i, p)$ *denotes the Euclidean distance between the center* c_i *and the point* p.

First we examine, how to solve the k-center problem for $k = 2$: We aim to construct two sequences of centers $m_{1,j}$ and $m_{2,k}$, based on two sequences of points from the two optimal balls B_1^* and B_2^* of equal radius.

We start with an arbitrary point p_1 and set $m_{1,1} = p_1$ as we can assume w. l. o. g. $p_1 \in B_1^*$. In every further step, we pick a point p_i farthest from either of the two current centers $c_{1,j}$ and $c_{2,k}$. So $p_i = \max_{p \in P} \min\{d(p, c_{1,j}), d(p, c_{2,k})\}$. As long as we have no center for B_2^*, we pick the point furthest from $c_{1,j}$. We then employ a guessing oracle that tells us whether p_i belongs to B_1^* or B_2^*. Depending on its answer, we add the point to the sequence for the respective ball then calculate a new center $m_{1,j+1}$ or $m_{2,k+1}$ as in our 1-center algorithm.

After $2\lfloor \frac{2}{\varepsilon} \rfloor + 1$ picks, at least one of the two sequences will contain $\lfloor \frac{2}{\varepsilon} \rfloor + 1$ many points, w. l. o. g. the sequence for B_1^*. Now consider the last point p_i that was added to the sequence of

B_1^*. By our arguments for the upper bound of rounds necessary for the 1-center algorithm, our center c_{i-1} for B_1^* based on the previous $\lfloor \frac{2}{\varepsilon} \rfloor$ many points already is a $(1+\varepsilon)$ approximation. Thus, $d(p_i, c_{i-1}) < (1+\varepsilon)r^*$. But p_i was picked as the point in P farthest away from either of the two current centers for B_1^* and B_2^*, thus, there exists no point in P with distance more than $(1+\varepsilon)r^*$ to either current center.

Thus, we obtain a $(1+\varepsilon)$-approximation within $2\lfloor \frac{2}{\varepsilon} \rfloor + 1$ rounds. As we don't have a guessing oracle, we just exhaust all possible guesses and return the best solution encountered, which results in a running time of $O\left(nd\,2^{\frac{1}{\varepsilon}}\right)$.

For $k > 2$, we apply the same approach by constructing k sequences of centers. By the same arguments as for $k = 2$, we arrive at a $(1+\varepsilon)$-approximation after at most $k\lfloor \frac{2}{\varepsilon} \rfloor + 1$ rounds. As we need to exhaust k possibilities each for these $k\lfloor \frac{2}{\varepsilon} \rfloor$ points, our algorithm runs in time $O\left(nd\,k^{\frac{k}{\varepsilon}}\right) = O\left(nd\,2^{\frac{k}{\varepsilon}\log k}\right)$.

Theorem 20

Given a set $P \in \mathbb{R}^d$, a $(1+\varepsilon)$-approximation of the Euclidean k-center problem can be computed in time $O\left(nd\,k^{\frac{k}{\varepsilon}}\right)$.

Chapter 5

Conclusion

Discrete Firefighting We introduced a new model for firefighting problems in graphs aimed to more realistically portray the spread of a fire using resistance and energy parameters. We showed that the r, e-model generalizes Hartnell's model, but allows to incorporate varied terrain parameters. We presented an algorithm to track fire propagation within a given graph, that is especially efficient in graphs with bounded maximum vertex degree, which includes most graphs used to model terrain.

We studied a number of problems focused on protecting a set of target cells at the boundary of a hexagonal cell graph from a spreading fire. When protecting them by adding an identical amount of resistance to a set of cells, we presented a polynomial time algorithm to decide, whether such a set of cells exists, while an algorithm to find such a set of minimum cardinality can run in exponential time in some graphs. When allowed to increase the resistance of cells individually, we provided an algorithm to solve this problem in polynomial time in graphs with restricted initial resistance and energy. As the results only apply to restricted classes of graphs, target sets or initial parameters, solutions for more general versions of this problem remain open. For the inverse problem of igniting a set of target vertices given a set of possible starter vertices, we showed NP-completeness.

Besides generalizing the algorithmic results for protection problem to more general graphs, we also discussed a number of other open problems, that allow for further research. One the one hand, the resistance and energy parameters allow to examine more nuanced versions of already well-studied problems in graphs, like the firefighter problem or the burning number. However, more unique features of our model, like the fire not necessarily igniting the whole graph, also allow for entirely new kinds of problems, e. g. the k-Evacuation Spot Problem.

In summary, the model presented succeeds as a framework with high flexibility and adaptability, both generalizing well-known problems as well as opening up new areas of research. However, as for the problems examined in this thesis, this flexibility can easily result in optimal solutions being too hard to compute in general instances. Instead, one should either restrict the problems to instances with specific structural properties, ideally derived from real-world instances, or strive to find good approximations.

Continuous Firefighting We have shown several non-trivial bounds for the speed necessary to contain the fire in the half-plane model. In addition we have shown simple strategies for other related problems like the angle cover or the infinite following fire model. Our results show that delaying barriers – in the half-plane model vertical segments attached to the horizontal barrier – can help to break the obvious upper bound of 2 for the building speed in some scenarios.

It remains the interesting question whether a similar effect could also be achieved for Bressan's original problem of containing the fire by a closed barrier curve. As an intermediate result in that direction, one ought to extend these results to more complex delaying barriers, e. g. those not necessarily connected to the horizontal barriers, as well as the Euclidean metric. However, as discussed in this thesis, the effects of delaying barriers are less pronounced and harder to analyse in both cases.

Minimum Enclosing Ball Problem We provided a new efficient gradient-descent $(1+\varepsilon)$ approximation algorithm for MEB in arbitrary dimensions running in time $O\left(\frac{nd}{\varepsilon}\right)$, which is strictly better than previous core-set based approaches with run times $O\left(\frac{nd}{\varepsilon}+\frac{1}{\varepsilon^{4.5}}\log\frac{1}{\varepsilon}\right)$ as long as $nd \in \Omega\left(\frac{1}{\varepsilon^{3.5}}\log\frac{1}{\varepsilon}\right)$. Like the core-set based algorithms it can be extended to the k-center problem with a run time of $O\left(ndk^{\frac{k}{\varepsilon}}\right)$. While this does not improve upon the most recent results by Rösner for input points with rational coefficients, the presented algorithm might nevertheless be particularly attractive due to its simplicity implying very low run-time constants.

As it is a generalization of a property already utilized in a streaming version of the Euclidean 2-center problem, it might be interesting to investigate, whether the structural property utilized in the algorithm might have applications in other similar clustering problems. Furthermore, the sequence of points picked by the algorithm is never as bad as the worst-case sequence we used to bound the necessary number of rounds. Thus, there might be room for improvement in the run-time analysis of our algorithm.

References

[1] Harold Abelson and Andrea A DiSessa. *Turtle geometry: The computer as a medium for exploring mathematics*. MIT press, 1986. doi:10.7551/mitpress/6933.001.0001.

[2] Elliot Anshelevich, Deeparnab Chakrabarty, Ameya Hate, and Chaitanya Swamy. Approximability of the firefighter problem. *Algorithmica*, 62(1-2):520–536, 2012. doi:10.1007/s00453-010-9469-y.

[3] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009. doi:10.1017/CBO9780511804090.

[4] J-P Aubin and Arrigo Cellina. *Differential inclusions: set-valued maps and viability theory*, volume 264. Springer Science & Business Media, 2012. doi:10.1007/978-3-642-69512-4.

[5] Franz Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405, 1991. doi:10.1145/116873.116880.

[6] Laécio C Barros, Rodney C Bassanezi, and Renata ZG De Oliveira. Fuzzy differential inclusion: An application to epidemiology. In *Soft Methodology and Random Information Systems*, pages 631–637. Springer, 2004. doi:10.1007/978-3-540-44465-7_78.

[7] Mihai Bădoiu and Kenneth L. Clarkson. Smaller core-sets for balls. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '03, page 801–802, USA, 2003. Society for Industrial and Applied Mathematics. doi:10.5555/644108.644240.

[8] Mihai Bădoiu and Kenneth L. Clarkson. Optimal core-sets for balls. *Computational Geometry*, 40(1):14 – 22, 2008. doi:10.1016/j.comgeo.2007.04.002.

[9] Mihai Bădoiu, Sariel Har-Peled, and Piotr Indyk. Approximate clustering via core-sets. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, STOC '02, page 250–257, New York, NY, USA, 2002. Association for Computing Machinery. doi:10.1145/509907.509947.

[10] Florian Berger, Alexander Gilbers, Ansgar Grüne, and Rolf Klein. How many lions are needed to clear a grid? *Algorithms*, 2(3):1069–1086, 2009. doi:10.3390/a2031069.

[11] Anthony Bonato. *The game of cops and robbers on graphs*. American Mathematical Soc., 2011. doi:10.1090/stml/061.

[12] Anthony Bonato, Jeannette Janssen, and Elham Roshanbin. How to burn a graph. *Internet Mathematics*, 12(1), 3 2016. doi:10.1080/15427951.2015.1103339.

[13] John Adrian Bondy, Uppaluri Siva Ramachandra Murty, et al. *Graph theory with applications*, volume 290. Macmillan London, 1976.

[14] Alberto Bressan. Differential inclusions and the control of forest fires. *Journal of Differential Equations*, 243(2):179–207, 2007. doi:10.1016/j.jde.2007.03.009.

[15] Alberto Bressan. Price offered for a dynamic blocking problem, 2011. URL http://personal.psu.edu/axb62/PSPDF/prize2.pdf.

[16] Alberto Bressan. Dynamic blocking problems for a model of fire propagation. In *Advances in Applied Mathematics, Modeling, and Computational Science*, pages 11–40. Springer, 2013. doi:10.1007/978-1-4614-5389-5_2.

[17] Alberto Bressan and Tao Wang. The minimum speed for a blocking problem on the half plane. *Journal of mathematical analysis and applications*, 356(1):133–144, 2009. doi:10.1016/j.jmaa.2009.02.039.

[18] Alberto Bressan and Tao Wang. Global optimality conditions for a dynamic blocking problem. *ESAIM: Control, Optimisation and Calculus of Variations*, 18(1):124–156, 2012. doi:10.1051/cocv/2010053.

[19] Alberto Bressan and Tao Wang. On the optimal strategy for an isotropic blocking problem. *Calculus of Variations and PDE*, 45:125–145, 2012. doi:10.1007/s00526-011-0453-4.

[20] Alberto Bressan and Dongmei Zhang. Control problems for a class of set valued evolutions. *Set-Valued and Variational Analysis*, 20(4):581–601, 2012. doi:10.1007/s11228-012-0204-5.

[21] Alberto Bressan, Maria Burago, Arthur Friend, and Jessica Jou. Blocking strategies for a fire control problem. *Analysis and Applications*, 6(03):229–246, 2008. doi:10.1142/S0219530508001146.

[22] Gerth Stølting Brodal, George Lagogiannis, and Robert E Tarjan. Strict fibonacci heaps. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 1177–1184, 2012. doi:10.1145/2213977.2214082.

[23] Leizhen Cai, Elad Verbin, and Lin Yang. Firefighting on trees:(1- 1/e)–approximation, fixed parameter tractability and a subexponential algorithm. In *International Symposium on Algorithms and Computation*, pages 258–269. Springer, 2008. doi:10.1007/978-3-540-92182-0_25.

[24] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 3rd Edition*. MIT Press, 2009. ISBN 978-0-262-03384-8.

[25] Mark de Berg, Marc van Kreveld, Otfried Schwarzkopf, and Mark Overmars. *Computational geometry: Algorithms and Applications*. Springer, 2000. doi:10.1007/978-3-540-77974-2.

[26] Mike Develin and Stephen G Hartke. Fire containment in grids of dimension three and higher. *Discrete Applied Mathematics*, 155(17):2257–2268, 2007. doi:10.1016/j.dam.2007.06.002.

[27] Zoltán Dezső and Albert-László Barabási. Halting viruses in scale-free networks. *Physical Review E*, 65(5):055103, 2002. doi:10.1103/PhysRevE.65.055103.

[28] Edsger W Dijkstra et al. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959. doi:10.1007/BF01386390.

[29] Zvi Drezner. The p-centre problem—heuristic and optimal algorithms. *Journal of the Operational Research Society*, 35(8):741–748, 1984. doi:10.2307/2581980.

[30] Markus Eiglsperger, Sándor P. Fekete, and Gunnar W. Klau. Orthogonal graph drawing. In *Drawing Graphs, Methods and Models.*, pages 121–171, 1999. doi:10.1007/3-540-44969-8_6.

[31] David Eppstein. Faster construction of planar two-centers. In *SODA*, volume 97, pages 131–138, 1997. doi:10.1145/314161.314198.

[32] Ben Evarts. Fire loss in the united states during 2018. *NFPA National Fire Protection Association, Quincy*, 2019.

[33] Stephen Finbow and Gary MacGillivray. The firefighter problem: a survey of results, directions and questions. *Australasian J. Combinatorics*, 43:57–78, 2009.

[34] Stephen Finbow, Andrew King, Gary MacGillivray, and Romeo Rizzi. The firefighter problem for graphs of maximum degree three. *Discrete Mathematics*, 307(16):2094–2105, 2007. doi:10.1016/j.disc.2005.12.053.

[35] Kaspar Fischer, Bernd Gärtner, and Martin Kutz. Fast smallest-enclosing-ball computation in high dimensions. In *European Symposium on Algorithms*, pages 630–641. Springer, 2003. doi:10.1007/978-3-540-39658-1_57.

[36] Patricia Fogarty. *Catching the fire on grids*. PhD thesis, University of Vermont, 2003.

[37] Bin Fu, Sorinel A Oprisan, and Lizhe Xu. Multi-directional width-bounded geometric separator and protein folding. *International Journal of Computational Geometry & Applications*, 18(05): 389–413, 2008. doi:10.1142/S0218195908002696.

[38] Martin Gardner. Mathematical games: The fantastic combinations of john conway's new solitaire game "life". *Scientific American*, 223:120–123, 1970.

[39] M. R. Garey and David S. Johnson. The rectilinear steiner tree problem in NP complete. *SIAM Journal of Applied Mathematics*, 32:826–834, 1977. doi:10.1137/S0097539704371353.

[40] Bernd Gärtner and Sven Schönherr. An efficient, exact, and generic quadratic programming solver for geometric optimization. In *Proceedings of the sixteenth annual symposium on Computational geometry*, pages 110–118, 2000. doi:10.1145/336154.336191.

[41] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science, 2nd Ed.* Addison-Wesley, 1994. ISBN 0-201-55802-5.

[42] Russell T Graham, Sarah McCaffrey, and Theresa B Jain. *Science basis for changing forest structure to modify wildfire behavior and severity*. United States Department of Agriculture Forest Service, Rocky Mountain Research Center, 2004. doi:10.2737/RMRS-GTR-120.

[43] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2. Springer Science & Business Media, 2012. doi:10.1007/978-3-642-78240-4.

[44] B.L. Hartnell. Firefighter! an application of domination. presentation. In *25th Manitoba Conference on Combinatorial Mathematics and Computing*. University of Manitoba in Winnipeg, Canada, 1995.

[45] RZ Hwang, Richard C. T. Lee, and RC Chang. The slab dividing approach to solve the euclidean p-center problem. *Algorithmica*, 9(1):1–22, 1993. doi:10.1007/BF01185335.

[46] David Kübel. *On some geometric search problems*. PhD thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, 2020.

[47] Sang-Sub Kim and Hee-Kap Ahn. An improved data stream algorithm for clustering. In Alberto Pardo and Alfredo Viola, editors, *LATIN 2014: Theoretical Informatics*, pages 273–284. Springer Berlin Heidelberg, 2014. doi:10.1007/978-3-642-54423-1_24.

[48] Sang-Sub Kim and Barbara Schwarzwald. A (1+ ε)-approximation for the minimum enclosing ball problem in $\mathbb{R}^d$. presentation, 2020.

[49] Sang-Sub Kim, Rolf Klein, David Kübel, Elmar Langetepe, and Barbara Schwarzwald. Geometric firefighting in the half-plane. In *Algorithms and Data Structures - 16th International Symposium, WADS 2019, Edmonton, AB, Canada, August 5-7, 2019, Proceedings, LNCS 11646*, pages 481–494, 2019. doi:10.1007/978-3-030-24766-9_35.

[50] Sang-Sub Kim, Rolf Klein, David Kübel, Elmar Langetepe, and Barbara Schwarzwald. Geometric firefighting in the half-plane. *CoRR*, abs/1905.02067, 2019.

[51] Andrew King and Gary MacGillivray. The firefighter problem for cubic graphs. *Discrete Mathematics*, 310(3):614–621, 2010. doi:10.1016/j.disc.2009.05.007.

[52] Rolf Klein. *Algorithmische Geometrie: Grundlagen, Methoden, Anwendungen*. Springer-Verlag, 2006. doi:10.1007/3-540-27619-X.

[53] Rolf Klein, Elmar Langetepe, and Christos Levcopoulos. A Fire Fighter's Problem. In Lars Arge and János Pach, editors, *31st International Symposium on Computational Geometry (SoCG 2015)*, volume 34 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 768–780. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015. doi:10.4230/LIPIcs.SOCG.2015.768.

[54] Rolf Klein, David Kübel, Elmar Langetepe, and Barbara Schwarzwald. Protecting a highway from fire. presentation, 2018.

[55] Rolf Klein, Christos Levcopoulos, and Andrzej Lingas. Approximation algorithms for the geometric firefighter and budget fence problems. *Algorithms*, 11(4):45, 2018. doi:10.1007/978-3-642-54423-1_23.

[56] Rolf Klein, David Kübel, Elmar Langetepe, Jörg-Rüdiger Sack, and Barbara Schwarzwald. A new model in firefighting theory. *CoRR*, abs/1911.10341, 2019.

[57] Rolf Klein, Elmar Langetepe, Barbara Schwarzwald, Christos Levcopoulos, and Andrzej Lingas. On a fire fighter's problem. *Int. J. Found. Comput. Sci.*, 30(2):231–246, 2019. doi:10.1142/S0129054119500023.

[58] Rolf Klein, David Kübel, Elmar Langetepe, Jörg-Rüdiger Sack, and Barbara Schwarzwald. A new model in firefighting theory. In *Conference on Algorithms and Discrete Applied Mathematics*, pages 371–383. Springer, 2020. doi:10.1007/978-3-030-39219-2_30.

[59] Ian Knight and John Coleman. A fire perimeter expansion algorithm-based on huygens wavelet propagation. *International Journal of Wildland Fire*, 3(2):73–84, 1993. doi:10.1071/WF9930073.

[60] Irina Kostitsyna, Maarten Löffler, Valentin Polishchuk, and Frank Staals. Most vital segment barriers. In *Workshop on Algorithms and Data Structures*, pages 495–509. Springer, 2019. doi:10.1007/978-3-030-24766-9_36.

[61] Piyush Kumar, Joseph S. B. Mitchell, and E. Alper Yildirim. Approximate minimum enclosing balls in high dimensions using core-sets. *J. Exp. Algorithmics*, 8:1.1–es, December 2004. ISSN 1084-6654. doi:10.1145/996546.996548.

[62] Richard J Lipton and Robert Endre Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979. doi:10.1137/0136016.

[63] Gary MacGillivray and Ping Wang. On the firefighter problem. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 47:83–96, 2003.

[64] Nimrod Megiddo. Linear-time algorithms for linear programming in $\mathbb{R}^3$ and related problems. *SIAM Journal on Computing*, 12(4):759–776, 1983. doi:10.1137/0212052.

[65] Nimrod Megiddo. On the complexity of some geometric problems in unbounded dimension. *Journal of Symbolic Computation*, 10(3):327 – 334, 1990. ISSN 0747-7171. doi:10.1016/S0747-7171(08)80067-3.

[66] Nimrod Megiddo and Kenneth J Supowit. On the complexity of some common geometric location problems. *SIAM journal on computing*, 13(1):182–196, 1984. doi:10.1137/0213014.

[67] Nimrod Megiddo, Arie Tamir, Eitan Zemel, and Ramaswamy Chandrasekaran. An $o(n\log^2 n)$ algorithm for the k^{th} longest path in a tree with applications to location problems. *SIAM Journal on Computing*, 10(2):328–337, 1981. doi:10.1137/0210023.

[68] Margaret-Ellen Messinger. Average firefighting on infinite grids. *Australasian Journal of Combinatorics*, 41:15, 2008.

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

[70] Richard Nowakowski and Peter Winkler. Vertex-to-vertex pursuit in a graph. *Discrete Mathematics*, 43(2-3):235–239, 1983. doi:10.1016/0012-365X(83)90160-7.

[71] Elsa Pastor, Luis Zárate, Eulalia Planas, and Josep Arnaldos. Mathematical models and calculation systems for the study of wildland fire behaviour. *Progress in Energy and Combustion Science*, 29(2):139–153, 2003. doi:10.1016/S0360-1285(03)00017-0.

[72] Romualdo Pastor-Satorras and Alessandro Vespignani. Epidemic spreading in scale-free networks. *Physical review letters*, 86(14):3200, 2001. doi:10.1103/PhysRevLett.86.3200.

[73] Paul Rendell. Turing universality of the game of life. In *Collision-based computing*, pages 513–539. Springer, 2002. doi:10.1007/978-1-4471-0129-1_18.

[74] Clemens Rösner. *Constrained Clustering Problems and Parity Games*. PhD thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, 2019.

[75] Raimund Seidel. Small-dimensional linear programming and convex hulls made easy. *Discrete & Computational Geometry*, 6(3):423–434, 1991. doi:10.1007/BF02574699.

[76] Alvy Ray Smith III. Simple computation-universal cellular spaces. *Journal of the ACM (JACM)*, 18(3):339–353, 1971. doi:10.1145/321650.321652.

[77] Xuehou Tan and Bo Jiang. Simple $o(n\log^2 n)$ algorithms for the planar 2-center problem. In *International Computing and Combinatorics Conference*, pages 481–491. Springer, 2017. doi:10.1007/978-3-319-62389-4_40.

[78] Guy Tennenholtz, Constantine Caramanis, and Shie Mannor. The stochastic firefighter problem, 2017.

[79] Tommaso Toffoli and Norman Margolus. *Cellular automata machines: a new environment for modeling*. MIT press, 1987.

[80] John Von Neumann. The general and logical theory of automata. *1951*, pages 1–41, 1951.

[81] Tao Wang. Optimality conditions for a blocking strategy involving delaying arcs. *Journal of Optimization Theory and Applications*, 152(2):307–333, 2012. doi:10.1007/s10957-011-9919-y.

[82] Emo Welzl. Smallest enclosing disks (balls and ellipsoids). In *New results and new trends in computer science*, pages 359–370. Springer, 1991. doi:10.1007/BFb0038202.

[83] S. Wolfram. *A new kind of science*. General science. Wolfram Media, 2002.

[84] Damián H Zanette and Marcelo Kuperman. Effects of immunization in small-world epidemics. *Physica A: Statistical Mechanics and its Applications*, 309(3-4):445–452, 2002. doi:10.1016/S0378-4371(02)00618-0.