Novel Developments in Physical and Computational Methods for Wildfire Modelling and Analysis

Conor Hackett BSc.

17338943

Hamilton Institute,

Maynooth University,

Maynooth, Co. Kildare,

Ireland.

A thesis presented in fulfilment of the requirements for the Degree of

Doctor of Philosophy

Supervisors: **Dr. Charles Markham** and **Dr. Rafael de Andrade Moral**

Initial Submission: 3rd July 2025

Viva Voce Examination: 6th October 2025

Final Submission: 9th January 2026

Table of Contents

Declaration

I hereby certify that this material, which I now submit for assessment on the program of study as part of the **Doctor of Philosophy** qualification, is *entirely* my own work and has not been taken from the work of others - save and to the extent that such work has been cited and acknowledged within the text of my work.

I hereby acknowledge and accept that this thesis may be distributed to future researchers, as an example of the standard expected of a PhD Thesis.

Signed: Conor Hackett Date: 09/01/2026

Acknowledgements

First, I would like to thank Charles Markham and Rafael de Andrade Moral for all their help and support over the past few years, culminating in this thesis. Without their excellent guidance and support this work would not have been possible (Figure 0.1).

I would also like to thank all the co-authors I worked with during my research journey including Michelle Doran, Denise Rooney, Gourav Misra, Tim Mc Carthy, Gabriel Palma and Ria Walsh. I would like to express my gratitude to John McDonald, Jonathan Yearsley, and Shirley O'Dea who were the internal examiner, external examiner and chair of my viva voce examination respectively. They provided valuable comments regarding the thesis. I would like to extend this thanks to all individuals that reviewed and provided feedback for academic papers submitted. I would also like to express my appreciation to all the members of the DecaMap team. They gave me a chance to join them in the role of modelling wildfires, where I got to meet many collaborators and take part in practice wildfire management exercises. This includes Stephanie Keogh who sourced a dataset of real burn scars in the Wicklow Mountains, Ireland that are used in this thesis.

I am incredibly grateful for the love and support my parents Cyril Hackett and Linda Hackett, my sister Ciara Hackett, my nanny Edith Dixon, my partner Aoife Flood and other family and friends showed me throughout this journey.

I would like to extend my gratitude to Janet Clifford, David Malone, Ken Duffy, Joanna O' Grady, Kate Moriarty, Rosemary Hunt, Patsy Finn and Peg Hanrahan from Research Ireland's CRT in the Foundations of Data Science and the Hamilton Institute, for organising CRT events and pointing me in the right direction to deal with Maynooth University's endless paperwork.

Finally, I would like to thank Fred Ameneyro, Eleni Zavrakli, Maeve Upton, Emma Govan, Blake Corrigan, Akash Satheesan, Yong Chen Goh, Pramit Dutta, Jonathan Giordano, Kevin Horan, Niloufar Sefidi, Jack Kilgallen, Dara MacConville, Solmaz Panahi, Victor Nagahama, Oluwayomi Akinfenwa, Anthony Gibbons, Thiago Cardoso, Brian O'Sulivan, Noah Otten,

Maira Rubab, Nahia Iturricastillo, André Menezes, Patrick Gorry, Ahmed Shalaby, Agustina Skiarski, Sean Mills, Daniel Nunez, Daniel Farcas, Luisa Presti, Samara Santos, Sabrina Marwede, Rachel McInerney, Enola Barvaeva, and Marina Antoniadou along with the remainder of the Hamilton Institute and CRT for giving me great advice and making the journey as enjoyable as possible.

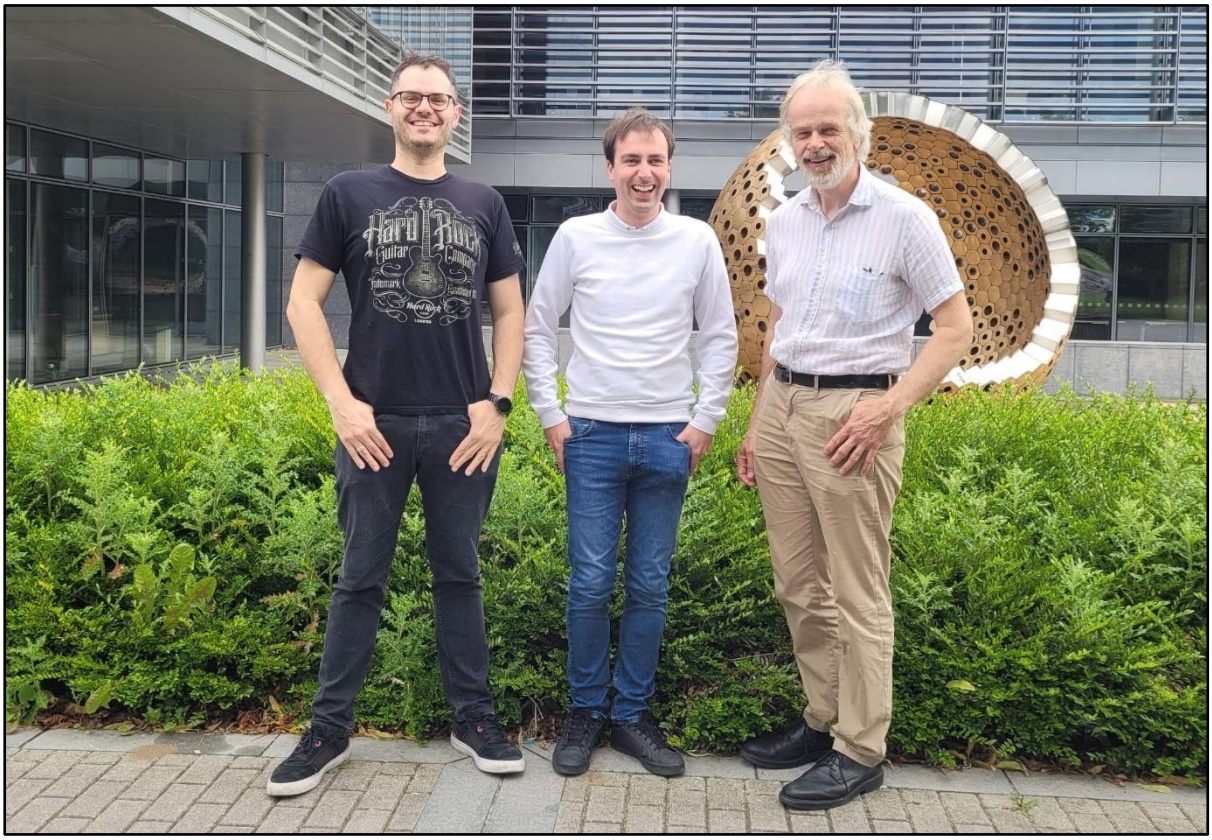

Figure 0.1: A photograph of Rafael de Andrade Moral (Left), Conor Hackett (Middle) and Charles Markham (Right) on the 20/06/2025. Image credit: Rachel McInerney.

Sponsor

This thesis has emanated from research conducted with the financial support of Taighde Éireann – Research Ireland under Grant numbers 18/CRT/6049 and SFI/21/FIP/DO/9945P. The opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Research Ireland.

Research Ireland Centre for Research Training
in Foundations of Data Science

Abstract

Wildfires are a natural disturbance which can cause harm to people, local ecology, the environment and infrastructure. Wildfires are theorised to have increased both in frequency and scale due to climate change. This has expanded interest in managing and reducing large scale, often catastrophic wildfires. A critical part of achieving this requires real-time fire line prediction. This provides insight as to how resources can be deployed to reduce the propagation of wildfires. Real-time fire line prediction is usually performed using a wildfire model. There are many frameworks wildfire models can simulate fire propagation with. A grid-based framework for modelling wildfires was developed in this thesis. Different types of grids were compared using this framework. A comparison was also made with a continuous-based framework. The use of a grid allowed the wildfire model to run fast. A critical research area regarding wildfire modelling often overlooked is the task of finding where a wildfire started and how long that wildfire burned. This information is important as it may help fire investigators determine the cause of the wildfire, which can be insightful for preventing future wildfires. A genetic algorithm which estimated both the wildfire starting locations and the propagation duration using its burn scar data was developed in this thesis. Typically, wildfire models are derived from experimental data. An investigation into the feasibility of using a desktop apparatus to model wildfires was completed in this thesis. The desktop apparatus was also used to validate the developed grid-based framework. A significant portion of this thesis was completed in collaboration with DecaMap, a group based in Maynooth University with the goal of developing emergency event management systems. This research was also completed with input from the Irish Fire Service and associated services. An Irish translation of the abstract is available on the next page.

Achoimre

Coiscriú nádúrtha iad na tinte fiáine, a dhéanann dochar don duine, as an damáiste don timpeallacht i gcoitinne idir ainmhithe agus fásra, agus infreastruchtúr. Deirtear go bhfuil borradh le líon réimse is minicíochta na dtinte fiáine de dheasca athraithe aeráide. Mar thoradh, tá suim méadaithe i ngnéithe láimhseála as an ísliú mór thinte fiáine, a bhíonn tubaisteach in amanna. Le beart mar seo a chur i gcrích is ríthábhachtach go mór réamhaisnéis líne tinte a bheith ann ar fhíor ama. Béarfar a leithéid de bheart léargas ar úsáid acmhainní chun líon na dtinte seo a laghdú. De ghnáth cuirtear i bhfeidhm fíor-ama réamhaisnéis tinte le do shamhail de thine a thástáil. Is féidir aithris a dhéanamh ar scaipeadh thinte leis an iomad creatlach de shamhlacha thinte fiáine. De bharr an tráchtais seo cruthaíodh creatlach de thinte fiáine bunaithe ar ghreillí. Cuireadh i gcomparáid greillí éagsúla bunaithe ar an chreatlach seo. Lena chois sin, cuireadh an ghreille i gcomparáid le creatlach leanúnach. Lig úsáid an ghreille do shamhail na tinte fiáine oibriú faoi luas. Réimse ríthábhachtach taighde maidir le samhaltú tinte fiáine a ndéantar faillí ann go minic is ea an tasc de theacht ar an áit inar thosaigh an tine agus cá fhad a mhair sé. Eolas an-tábhachtach atá ann óir féadfaidh sé chuidiú le lucht fiosraithe cúiseanna tinte fiáine a nochtadh as an gcúnamh a thabhairt chun tinte a chosc amach anseo. De thoradh an tráchtais seo tá algartam géiniteach anois ann a bhfuil in ann na pointí a thosaigh an tine fhiáin agus a scaipeadh tréimhse de réir sonraí lorg dóite. De ghnáth, gabhtar samhail de thinte as sonraí trialacha. Sa tráchtas seo rinneadh iniúchadh ar fhéidearthachtaí feidhm a bhaint as uirlis deisce chun samhail a dhéanamh ar thinte fiáine. Baineadh feidhm as an uirlis deisce fosta chun creatlach bunaithe ar ghreille a dheimhniú. Rinneadh an-chuid den tráchtas seo i gcomhar le DecaMap, grúpa atá lonnaithe in Ollscoil Mhá Nuad ar chuspóir dóibh córas bainistíochta eachtraí éigeandála a fhorbairt. Rinneadh an taighde seo le hionchur ó Sheirbhísí Dóiteáin na hÉireann agus seirbhísí a bhaineann léi. Tá leagan Béarla den achomaireacht seo ar fáil ar an leathanach roimhe.

List of Figures

List of Tables

List of Acronyms

The acronyms use in this thesis are described in Table 0.1.

Acronym	Definition
CRT	Centre for Research Training
USA	United States of America
SI	International System of Units
FBP	Canadian Forest Fire Behavior Prediction System
CNN	Convolutional Neural Network
UAV	Unmanned Aerial Vehicle
GIS	Geographic Information System
EPSG	European Petroleum Survey Group
SNAP	Sentinel Application Platform
JSON	JavaScript Object Notation
KML	Keyhole Markup Language
ML	Machine Learning
YOLO	You Only Look Once
RGB	Red, Green and Blue
HSV	Hue, Saturation and Value
GA	Genetic Algorithm
cWGAN	Conditional Wasserstein Generative Adversarial Network
WRF–SFIRE	Weather Research and Forecasting-Wildfire Spread Simulation
IGS	Irregular Grid Software
Eq.	Equation
D	Dimensional
API	Application Programming Interface
NetCDF	Network Common Data Form
ROC	Receiver Operating Characteristics
CORINE	Coordination of Information on the Environment
WSGA	Wildfire Source Genetic Algorithm
GPU	Graphics Processing Unit
ID	Identification
HD	High Definition
CPU	Central Processing Unit
PMW	Pulse Width Modulation
SD	Standard Deviation
SE	Standard Error
NCV	Net Calorific Value
COP	Common Operational Picture
SIM	Subscriber Identity Module

ADV	Aerial Drone Video
Geo-ML	Geocoded Machine Learning
EU	European Union
MODEX	EU Module Exercises
RI	Research Ireland (formerly Science Foundation Ireland)
SFI	Science Foundation Ireland (now Research Ireland)
RASM	Realtime Aera Surveillance & Mapping
RTÉ	Raidió Teilifís Éireann
CS	Computer Science
CSO	Central Statistics Office
CASI	Conference on Applied Statistics in Ireland
EA	Enterprise Alliance
Young-ISA	Young-Irish Statisticians Association
TSE	Theoretical and Statistical Ecology Group
ESA	European Space Agency
DOS	Dark Object Subtraction
NDVI	Normalised Difference Vegetation Index
NDBI	Normalised Difference Build Up Index
NDTI	Normalised Difference Tillage Index
SWIR	Short-Wave Infrared
NIR	Near Infrared

Table 0.1: List of acronyms and associated definitions used in this thesis.

Chapter 1 Introduction

This chapter starts with an examination of the impact of wildfires on Earth. The importance of modelling wildfires to increase understanding in this area is explained. The different techniques of modelling and simulating wildfires are then introduced. The remainder of the chapter explores open problems in wildfire modelling. The chapter ends with a guide to the remaining chapters of the thesis.

1.1 The Impact of Wildfires and Climate Change

A Wildfire is a potentially destructive fire that burns throughout an area; this includes forest fires and bushfires [1]. Usually, they begin in remote areas where there is a higher density of combustible vegetation such as grasslands and peatlands. They can sometimes threaten urban areas too [2]. Wildfires can be caused by both natural events such as lightning and human-caused actions such as arson and farming techniques like slash-and-burn agriculture [3] [4]. The frequency of wildfires has increased within forested extratropical and boreal regions in recent years; this is most likely due to climate change creating drier terrain, allowing for fires to burn more easily [5] [6]. However, this varies regionally, where factors such as the wildland-urban interface, land use changes, and fire exclusion policies can have an impact on the number of wildfires [7].

Climate change and land-use changes are projected to make wildfires more frequent and intense, with a global increase in the number of extreme fires of up to 14% by 2030, 27% by the end of 2050 and 57% by 2100 [8]. These values were forecasted by a Representative Concentration Pathway with a radiative forcing value of $6W/m^2$ which represents the energy imbalance in Earth's energy system caused by greenhouse gases and other factors, in the year 2100.

Areas of high elevation and dense tree cover tend to be more susceptible to lightning strikes which can start wildfires [9]. There is typically no temporal pattern of lightning ground flashes [10]. Due to their elevation, these areas tend to have less flora and therefore less

fuel that can burn [11]. However, boreal forests can grow at high altitudes [12]. The Northeastern boreal forests in China are an example of this [13]. It is important to note that in the past it was possible that wildfires started by lightning may have helped reduce forest litter, preventing larger wildfires. This parallels a tactic used by the Aboriginal people in Australia where they performed controlled burning to prevent large scale bushfires [14].

There are three primary types of wildfires: underground, surface, and tree crown (canopy) wildfires. Underground wildfires typically occur in organic matter underground such as plant roots or peat [15]. These wildfires tend to be a slowly propagating, smouldering fire that may persist for a long time. Surfaces wildfires are usually a flaming fire that use pine needles, grass, shrubs and forest littler on the ground as fuel [16]. Tree crown fires tend to be flaming fires that occur on the branches and leaves of alive, upright trees [16]. Their propagation is dependent on the amount and continuity of trees in the area. They are more likely to cause wildfire spotting due to their increased heigh. Wildfire spotting is the creation of additional fires separate from the main wildfire as ignited fuel from the main wildfire falls into new combustible vegetation [17].

The devastation of wildfires can be measured in different ways such as fatalities, ecological damage and economic damage. In California, United States of America (USA), and Australia a small subset of wildfires that burn in an environment satisfying the correct conditions can potentially grow to be very large fast-burning fires which can present an increased risk to life [18] [19]. They are also capable of causing negative long term health effects in humans [20] [21]. Wildfires impact the ecosystem by destroying some of the flora and displacing or killing the fauna [22]. Annually wildfires destroy approximately 350 million hectares of forest globally [23]. When wildfires burn through terrain, they emit greenhouse gasses such as carbon dioxide, further contributing climate change [24] . The economic impact of wildfires includes damage to the agricultural and forestry sectors [25]. When a wildfire spreads to an urban area it can also cause considerable damage to town infrastructure [2]. Local economies have been damaged in the past by wildfires especially in areas where wildfires are a common occurrence such as California, USA [26]. Due to the risks that wildfires pose, it is a high priority for countries around the world to prevent or contain them [27].

Countries that normally have a wet climate, such as Ireland, may also start to have an increased number of wildfires due to warmer temperatures, likely caused by climate change [28] [29]. Wildfires have had an ecological impact in Ireland where they tend to be smaller, slower-burning fires occurring mainly in bogs (peatlands), which are home to many rare species of both flora and fauna [30]. The high abundance of gorse (*Ulex europaeus*) in fens and common heather (*Calluna vulgaris*) in bogs makes them frequent places for wildfires to occur as these plants are easily ignitable [31] [32] [33].

Ecological diversity has an impact on wildfire scale, as some types of grass may increase the propagation rate of wildfires due to their fine fuel loads such as the invasive species Giant Reed (*Arundo donax*) in the USA [34]. However, It is important to note that the outcome of a wildfire is not always completely negative as pyrophytic plants species such as the Aleppo pine (*Pinus halepensis*) benefit from regularly occurring wildfires [23] [35]. Its seed which is enclosed in pinecones only escapes when the pinecones are heated to a high enough temperature, melting their resin and releasing the seed, thus giving the seeds a chance to germinate.

There is a large variety of burning behaviours associated with different types of wildfires found around the world. With the correct parametrisation and physical understanding, these fires can all be modelled. Once challenge is estimating the parameters characterising fuel properties, terrain and weather. The work completed in this thesis addresses aspects of this challenge and was completed with feedback from professionals who manage wildfires.

1.2 Modelling and Simulating Wildfires

With the increasing severity and frequency of wildfires, the ability to model and predict wildfire propagation has become an invaluable asset to planners and firefighters. Chief Fire Officers can use this information to forecast and prevent fires, while firefighters can also use this information to find the optimal locations to apply interventions [36].

There are three main branches of wildfire modelling: statistical, empirical, and physical models [37]. Some wildfire models may contain aspects of more than one branch of wildfire modelling. Statistical models are built on a statistical description of wildfires found by observing sample fires. These models are less focused on the thermodynamics of fire and use statistics to estimate the propagation rate of a wildfire. The McArthur Forest Fire Danger Index is a statistical wildfire model [38] [39]. The model takes variables such as the temperature of curing for the fuel, air temperature, relative humidity, and wind velocity to produce a value representing the fire danger index. This value can then be converted into an estimated rate of spread for a wildfire. This statistical model was developed by only sampling fires burning dry grassland and forest litter which means it needs to be used with caution on other fuel types, as that would require extrapolation.

Empirical wildfire spread models are constructed using observed propagation behaviour. They focus on how a wildfire would propagate given a set of conditions. An empirical wildfire model was built using observed wildfire data in Northwestern Europe [40]. The median observed propagation rate of wildfires on different types of land cover was recorded. These values can then be used to model future wildfires in the analysed land cover types.

Physical wildfire propagation models are built from the underlying physics that causes wildfires to propagate. In [41] the propagation rate of wildfires in porous fuel beds was found. This model calculated the rate of spread by finding the heat generated from the combustion of existing fuel and spreading this heat using the principles of the conservation of energy into the surrounding fuel, until that fuel reaches its flash point and ignites, continuing the combustion process. Another physical model was developed to find the rate of spread between two particles in a fuel bed [42]. This model found the time it took for fire to spread the distance from one particle to another to calculate rate of spread.

All three branches of wildfire modelling are explored in thesis.

1.3 Open Problems

The identification of open problems was achieved by examining the research area, reviewing relevant literature (Chapter 2), and from discussions with professional firefighters and active researchers.

The first open problem identified was to demonstrate integration of wildfire models into a live, interactive web service that would help fire services plan for wildfire exercise events. A core part of this integration was engagement with groups who manage wildfires including the Irish Fire Service, Irish Air Corps, National Parks and Wildlife Services, and Coillte during development. This is part of a challenge-based project discussed in Chapter 6 [43].

Many of the current frameworks for wildfire modelling have potential to be improved in terms of computational efficiency and precision. Grid-based frameworks including irregular grids provide a platform where there is an appropriate balance of both computational efficiency and precision. In Chapter 3 an irregular grid-based framework was developed, and different grid types were explored. Increased computational efficiency allows multiple wildfires to be simulated in quick succession. Improvements in efficiency opens other applications such as solving inverse problems in wildfire modelling.

Locating the ignition point of a burn scar is an area of research with minimal literature. An inverse approach was developed to address this challenge described in Chapter 4. This included automating the estimation of wildfire ignition points and propagation durations. It used the irregular grid-based framework due to its discrete spatial nature and increased computational efficiency.

Most wildfire models are built using a dataset containing information regarding wildfire propagation under different conditions [44]. The collection of real wildfire data is typically challenging. A study to investigate the feasibility of using small fires to validate larger models was considered in Chapter 5. These laboratory experiments have utility for educational and demonstration purposes but also have potential to inform changes in wildfire models and provide experimental validation for wildfire frameworks developed.

1.4 Thesis Guide

This thesis explores the area of wildfire modelling. It makes three main contributions to the area: 1. a study of grid-based models, 2. solving the inverse problem of relating burn scar data to ignition points, 3. and a desktop apparatus for collecting burn data for computer models. Chapter 2 provides a literature review that guided the research. The papers identified, support the work done on the open problems that were investigated. In Chapter 3, a new framework for modelling wildfires on a grid was developed. This includes a comparison of different grid types (triangular, square, hexagonal, random with an even distribution and random with areas of focus). Chapter 4 describes a novel method to estimate wildfire ignition points and propagation durations from burn scars using genetic algorithms. Chapter 5 compares the framework developed in Chapter 3 to experimental data collected from burns on a desktop apparatus designed capture key parameters relating to wildfire models (including slope and wind speed). In Chapter 6 wildfire exercises and professional events that were attended are described along with the tools created that were presented at them. The tools developed were used to inform the timeline and sequence of events for the exercises. The report reflects on the importance on development of effective tools. The professionals interacted with in Chapter 6 provided rationale for the research completed in Chapter 3 to Chapter 5. Chapter 7 provides a reflection on the research in this thesis and incudes a discussion on how it could be expanded in the future.

Chapter 2 Background and Theoretical Foundations

This chapter outlines the background work that inspired the contributions in this thesis (Figure 2.1). It begins with a review of existing wildfire models and their relevance to the overall work. Next the two types of frameworks to model wildfires are explained, continuous space and cellular automata. Previously developed frameworks and their connection to the thesis are also discussed. Physical wildfire models can be categorised based on fuel bed size as, small, medium and large. Existing studies in physical wildfire modelling using these fuel bed sizes are examined. A potentially unique feature of the work presented in this thesis is the level of engagement with stakeholders. These interactions informed the design of the artefacts created. The information learned from these professionals in their field is discussed. This information includes what groups manage wildfires in Ireland and how they do it. There is also a segment describing how fire investigators find wildfire ignition points manually. The following section discusses how Geographic Information System data were used in this thesis. This included satellite imagery which functioned as a data source and Geographic Information System features which stored outputs from simulations in a format readable by other analysis programs. The next section categorises computer vision into machine learning and traditional approaches. Both types and their relevance to the work in the thesis are discussed. The following section describes different approaches to solving inverse problems. Their advantages and disadvantages are examined. There is a detailed segment on genetic algorithms, which are a technique used later in the thesis. The final sections provides a recap of the points discussed throughout this chapter.

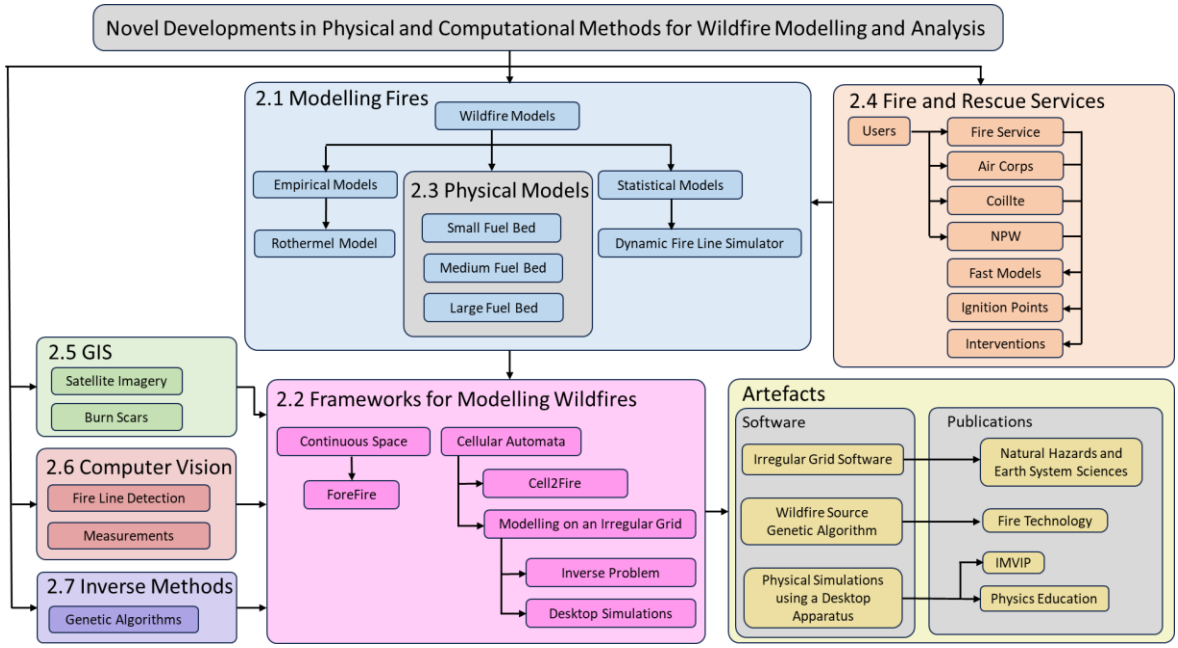

Figure 2.1: An overview of topics discussed in the literature review. This diagram also contains artefacts created in the thesis.

2.1 Modelling Fires

This section begins with an explanation of the science of combustion, followed by a detailed description of the Rothermel model, a summary of the Canadian Forest Fire Behavior Prediction System and a brief section regarding other approaches. Physical wildfire models are explored in Section 2.3.

2.1.1 The Science of Combustion

All fires, including wildfires, require four components to support combustion [45]. These are fuel (vegetation), an oxidising agent (oxygen in the air), heat (initiates the process) and a chemical chain reaction (to sustain combustion). Combustion is a chemical process in which fuel and an oxidiser combine generating heat greater than the fuel-oxygen mixture's flash point; if there is sufficient heat generated to overcome heat loss, combustion is sustained in a chain reaction. Together they make up the "fire tetrahedron"; a fire will cease to exist if any component is diminished within the tetrahedron (Figure 2.2). The applications of this will be explored in Chapter 6.

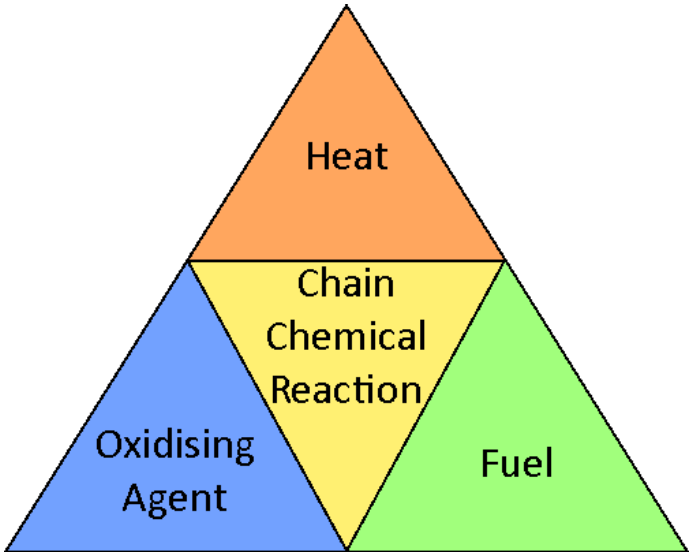

Figure 2.2: A 2D net of the fire tetrahedron, which shows all four components required to support combustion.

2.1.2 The Rothermel Model

One of the most cited wildfire models is the Rothermel model [46] [47]. It is used in the following chapters. The Rothermel model was developed by Richard C. Rothermel in 1972 to help forest managers predict the behaviour of surface wildfires. The Rothermel model represents one approach to wildfire modelling as a physics-based, semi-empirical model formed using the conservation of energy that underpins many operational fire-modelling tools. The Rothermel model is based on the laws of heat transfer but was developed using experimental data from constructed fuel beds.

The Rothermel model takes wind, terrain and fuel bed factors as input to produce an estimated rate of spread as an output (Eq. 2.1). The numerator measures total heat transferred to neighbouring fuel, while the denominator measures energy required to ignite neighbouring fuel. R is the spread rate of the fire. Rate of spread is measured on a 1-D line. This can be expanded to additional dimensions using wildfire modelling frameworks which include measuring R on separate lines along each direction of interest. An easy way to interpret this is a section of burning terrain acts as a heat source, while the fuel and fuel bed of the neighbouring terrain acts as a heat sink. The surplus heat produced will ignite the neighbouring area. The size of the ignited area is determined by the amount of surplus heat

(Figure 2.3). The Rothermel model is normally presented using United States customary units. In this thesis these values were converted into SI units before and after all wildfire model calculations. Another approach that could be considered would be to use the Rothermel equation reformulated in SI units [48]. This approach was not taken to make later comparisons in the thesis fair (Section 2.2.2) (Chapter 3). The original Rothermel equation is written as:

$$R = \frac{I_R \xi (1 + \phi_w + \phi_s)}{\rho_b \varepsilon Q_{ig}},$$

(Eq. 2.1)

where R is the rate of spread in m/s (ft/min), I_R is the reaction intensity in J/m^2/s (Btu/ft^2/min), ξ is the dimensionless propagation flux ratio, ϕ_w is the dimensionless wind factor, ϕ_s is the dimensionless slope factor, ρ_b is the oven-dry fuel bulk density in kg/m^3 (lb/ft^3), ε is the dimensionless effective heating number, and Q_{ig} is the heat of preignition in J/kg (Btu/lb). The variables in Section 2.1.2 are explained in Table 2.1. The environmental factors (input) produce a rate of spread using the Rothermel model. The model consists of a series of linked equations (Figure 2.4). These equations are also listed in [47] and [46].

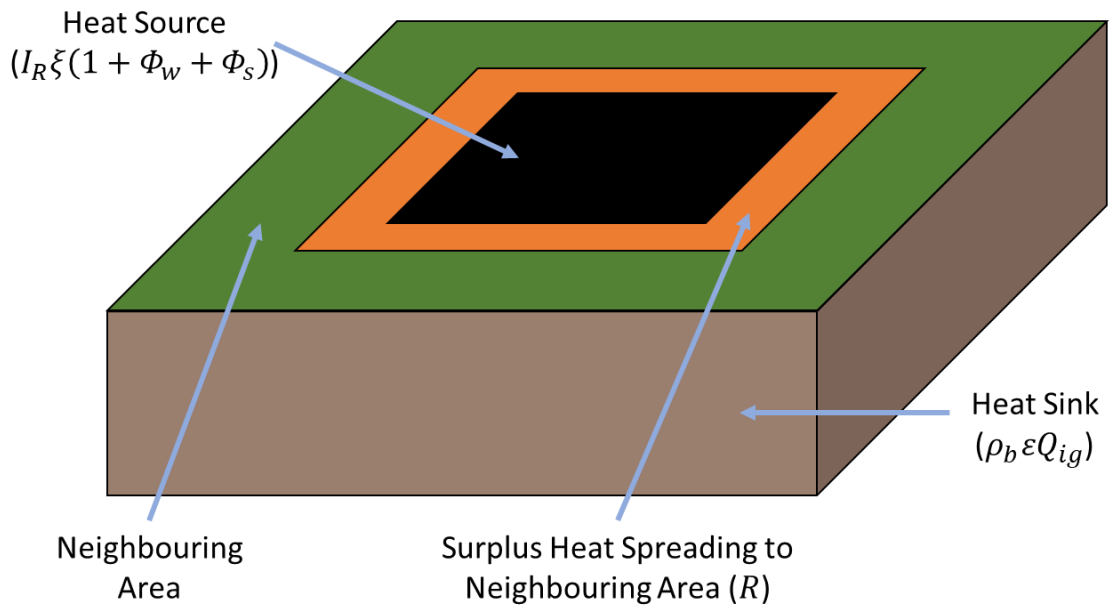

Figure 2.3: Example of how the Rothermel model functions, where the heat source is the location of the fire (where heat is being generated from combustion); the heat sink absorbs heat from the heat source, slowing down propagation; and surplus heat is used to spread fire to the neighbouring area.

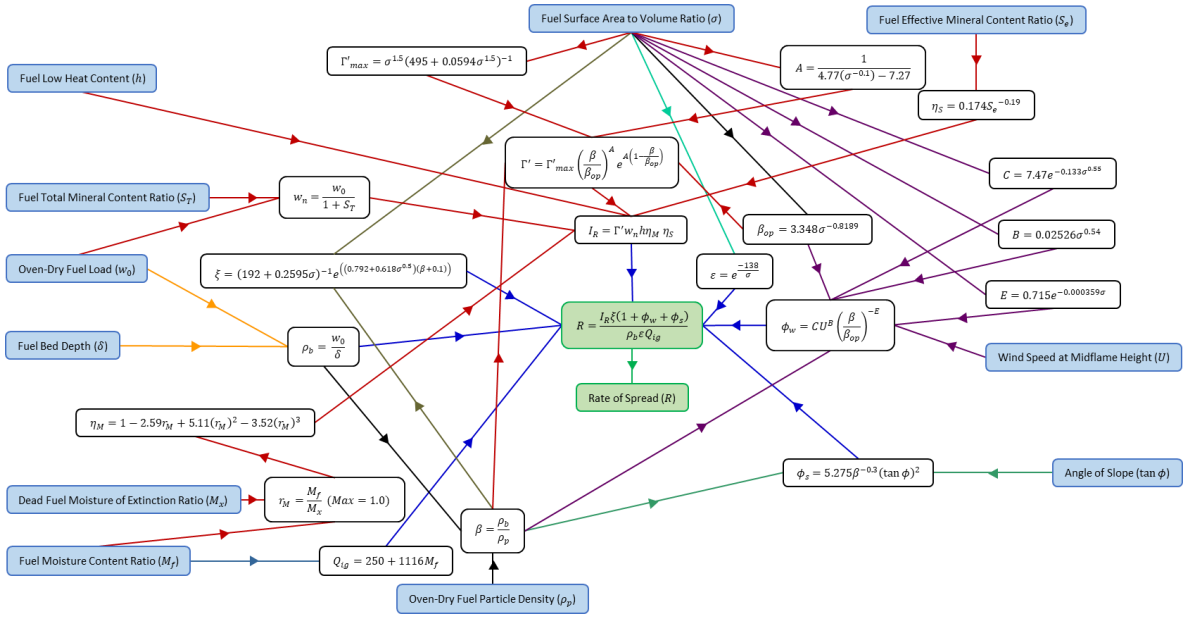

Figure 2.4: A description of the linked equations and inputs of the Rothermel model producing a rate of spread. Boxes with a blue background are input variables. The Rothermel equation and rate of spread are represented by boxes with a green background. The lines connecting boxes are coloured according to what variable of the Rothermel model they influence (dark blue arrows represent variables that appear in the Rothermel equation, black arrows represent variables that affect multiple values in the Rothermel equation, and the green arrow represents the final outputted rate of spread).

The oven-dry bulk density (ρ_b) is the density of total fuel mass in the fuel bed, in kg/m^3 (lb/ft^3). This value is described by dividing the oven-dry fuel load by the fuel bed depth as follows:

$$\rho_b = \frac{w_0}{\delta},$$

(Eq. 2.2)

where w_0 is the oven-dry fuel load in m^2 (ft^2) and δ is the fuel bed depth in m (ft).

The packing ratio β is a dimensionless ratio of pure fuel in the fuel bed. It is described as follows:

$$\beta = \frac{\rho_b}{\rho_p},$$

(Eq. 2.3)

where ρ_p is the oven-dry fuel particle density in kg/m^3 (lb/ft^3).

The optimum packing ratio (β_{op}) is the dimensionless packing ratio required for the maximum reaction intensity (I_R), in J/m^2/s (Btu/ft^2/min). It is calculated using experimental data to form the equation as follows:

$$\beta_{op} = 3.348\sigma^{-0.8189},$$

(Eq. 2.4)

where σ is the fuel surface area to volume ratio in m (ft).

The moisture damping coefficient (η_M) is a dimensionless value which slows down the reaction intensity (I_R) of the fire due to moisture. It is calculated using experimental data to form the following equations:

$$\eta_M = 1 - 2.59r_M + 5.11(r_M)^2 - 3.52(r_M)^3,$$

(Eq. 2.5)

$$r_M = \frac{M_f}{M_x},$$

(Eq. 2.6)

where r_M is a dimensionless placeholder variable for calculations (please note that r_M has a maximum value of 1.0), M_f is the dimensionless fuel moisture ratio and M_x is the dimensionless dead fuel moisture of extinction coefficient.

The mineral damping coefficient (η_S) is a dimensionless value which slows down the reaction intensity (I_R) of the fire due to minerals. It is calculated using experimental data to form the equation as follows:

$$\eta_S = 0.174{S_e}^{-0.19},$$

(Eq. 2.7)

where S_e is the dimensionless effective fuel mineral content ratio (no silica).

The net fuel load (w_n) is the oven-dry fuel load, excluding the mass of minerals, in kg/m^2 (lb/ft^2). It is described as follows:

$$w_n = \frac{w_0}{1 + S_T},$$

(Eq. 2.8)

where S_T is the dimensionless total mineral content ratio.

The maximum reaction velocity (Γ'_{max}) is the highest possible reaction velocity for a fuel given its surface area to volume ratio, in s^{-1} (min^{-1}). It is calculated using experimental data to form the equation as follows:

$$\Gamma'_{max} = \sigma^{1.5}(495 + 0.0594\sigma^{1.5})^{-1}.$$

(Eq. 2.9)

The optimum reaction velocity (Γ') is the velocity of a fuel's reaction in the absence of moisture and minerals, in s^{-1} (min^{-1}). It is calculated using experimental data to form the following equations:

$$\Gamma' = \Gamma'_{max}\left(\frac{\beta}{\beta_{op}}\right)^{A} e^{A\left(1-\frac{\beta}{\beta_{op}}\right)},$$

(Eq. 2.10)

$$A = \frac{1}{4.77(\sigma^{-0.1}) - 7.27},$$

(Eq. 2.11)

where Γ' is the optimum reaction velocity in s^{-1} (min^{-1}), and A is a dimensionless placeholder variable for calculations.

The reaction intensity (I_R) is the amount of energy released per unit area of the fire front per unit time, in $J/m^2/s$ ($Btu/ft^2/min$). It is calculated as follows:

$$I_R = \Gamma' w_n h \eta_M \eta_S,$$

(Eq. 2.12)

where h is the fuel low heat content (also known as net calorific value or lower heating value) in J/kg (Btu/lb).

The propagation flux ratio (ξ) is the dimensionless proportion of heat energy from the combustion reaction transferred to adjacent fuel particles. It is calculated using experimental data to form the equation as follows:

$$\xi = (192 + 0.2595\sigma)^{-1} e^{\left((0.792+0.618\sigma^{0.5})(\beta+0.1)\right)}.$$

(Eq. 2.13)

The effective heating number (ε) is a dimensionless value ranging between 0 and 1 which represents the ratio of fuel particles in a fuel bed that were heated to their flash points before the fire reached them. It is calculated using experimental data to form the equation as follows:

$$\varepsilon = e^{\frac{-138}{\sigma}}.$$

(Eq. 2.14)

The heat of preignition (Q_{ig}) is the energy per unit mass required for fuel to ignite, in J/kg (Btu/lb). The value 250 Btu/lb represents the amount the amount of energy required for the fuel to ignite. This value is sourced by multiplying the specific heat capacity of cellulose by the flash point of cellulose which produces a value of 581,500 J/kg (250 Btu/lb). The value 1,116 Btu/lb represents the latent heat of vaporisation of water which is 2,595,816 J/kg (1,116 Btu/lb). The heat of preignition is calculated as follows:

$$Q_{ig} = 250 + 1116M_f.$$

(Eq. 2.15)

The slope factor (ϕ_s) is a dimensionless value which increases the propagation rate of the fire based on the steepness of the slope. It is calculated using experimental data to form the equation as follows:

$$\phi_s = 5.275\beta^{-0.3}(\tan(\phi))^2,$$

(Eq. 2.16)

where $\tan(\phi)$ is the angle of slope in °D.

The wind factor (ϕ_w) is a dimensionless value which increased the propagation rate of the fire based on the velocity of wind until an upper limit. The wind factor is calculated using experimental data to form the following equations:

$$\phi_w = CU^B \left(\frac{\beta}{\beta_{op}}\right)^{-E},$$

(Eq. 2.17)

$$U_{max} = 0.9I_R,$$

(Eq. 2.18)

$$B = 0.02526\sigma^{0.54},$$

(Eq. 2.19)

$$C = 7.47e^{-0.133\sigma^{0.55}},$$

(Eq. 2.20)

$$E = 0.715e^{-0.000359\sigma},$$

(Eq. 2.21)

where U is the wind speed at midflame height in m/s (ft/min), U_{max} is the wind limit in m/s (ft/min) (please note that U has a maximum value of U_{max}), the variables B, C and E are dimensionless placeholder variables for calculations.

Variable	Units	Definition
R	m/s (ft/min)	Rate of spread.
I_R	J/m^2/s (Btu/ft^2/min)	Reaction intensity.
ξ	Dimensionless	Propagation flux ratio.
ϕ_w	Dimensionless	Wind factor.
ϕ_s	Dimensionless	Slope Factor.
ρ_b	kg/m^3 (lb/ft^3)	Oven-dry fuel bulk density.
ε	Dimensionless	Effective heating number.
Q_{ig}	J/kg (Btu/lb)	Heat of preignition.
w_0	m^2 (ft^2)	Oven-dry fuel load.
δ	m (ft)	Fuel bed depth.
β	Dimensionless	Packing ratio.
ρ_p	kg/m^3 (lb/ft^3)	Oven-dry fuel particle density.
β_{op}	Dimensionless	Optimum packing ratio.
σ	m (ft)	Fuel surface area to volume ratio.
η_M	Dimensionless	Moisture damping coefficient
r_M	Dimensionless	Placeholder variable for calculations.
M_f	Dimensionless	Fuel moisture ratio.
M_x	Dimensionless	Dead fuel moisture of extinction ratio.
η_S	Dimensionless	Mineral damping coefficient.
S_e	Dimensionless	Effective fuel mineral content ratio (no silica).
w_n	kg/m^2 (lb/ft^2)	Net fuel load.
S_T	Dimensionless	Total fuel mineral content ratio.
Γ'_{max}	s^{-1} (min^{-1})	Maximum reaction velocity.
Γ'	s^{-1} (min^{-1})	Optimum reaction velocity.
A	Dimensionless	Placeholder variable for calculations.
h	J/kg (Btu/lb)	Fuel low heat content (net calorific value or lower heating value).
$\tan(\phi)$	°D	Angle of slope.
U	m/s (ft/min)	Wind speed at midflame height.
U_{max}	m/s (ft/min)	The wind limit.
B	Dimensionless	Placeholder variable for calculations.
C	Dimensionless	Placeholder variable for calculations.
E	Dimensionless	Placeholder variable for calculations.

Table 2.1: List of variables, and their associated definition used in the Rothermel model.

Aligning the literature descriptions with the existing online code repositories was a significant challenge [46] [47] [49]. It is by analysing each of these separate equations that the Rothermel model used was brought into agreement with its existing implementations.

2.1.3 Canadian Forest Fire Behavior Prediction System (Model)

The Canadian Forest Fire Behavior Prediction System (FBP) is another well cited wildfire model developed by Kevin Hirsch in 1996 [44]. It is a viable alternative to the Rothermel model. It models both surface and tree crown wildfires using 14 different inputs (Figure 2.5). The model was not used in this thesis as a dataset containing fuel properties for Europe was not as readily available as it was for the Rothermel model. It is however helpful to review it as it highlights some of the strengths and weaknesses of the Rothermel model.

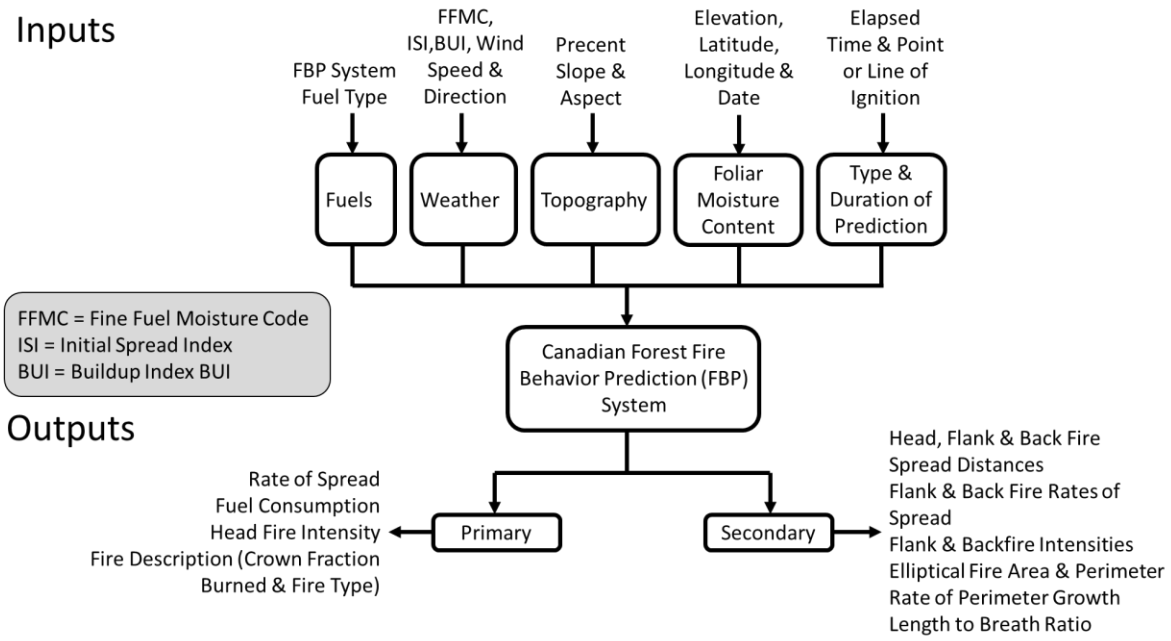

Figure 2.5: Canadian Forest Fire Behavior Prediction System (FBP) pipeline. Image based on FBP System's User Guide [44].

The first input is type of land cover (fuel). When initially published the FBP used 16 types of land cover from Canada. The next input is fine fuel moisture content ratio which is a dimensionless value that had to be measured. The following input is a dimensionless value called the initial spread index. One method of calculating the initial spread index is from a table of values based on fine fuel moisture content ratio and wind speed. The build-up index is a dimensionless input describing the amount of fuel available for combustion. This value is calculated using a numerical rating of the average moisture content of loosely compact organic layers at moderate depth (up to 0.0762m) and a numerical rating of the average moisture content of compact organic layers deep in fuel bed (deeper than 0.0762m). The next two inputs are wind speed (km/h) and wind direction (°D). The angle of

slope (°D) and slope direction (aspect) (°D) are also inputs. The base fuel moisture content is found using the following spatial-temporal data as inputs: longitude (°D), latitude (°D), elevation (m) and date. The propagation duration of the wildfire and either the position of ignition points or an area that is currently on fire are the final two inputs for the FBP.

The model outputs a rate of spread in the direction of the wind called the head fire rate. The FBP generates wildfires in a common elliptical pattern which also requires information regarding the back fire rate and flank fire rate. The back fire is the rate at which the fire moves against the wind. This rate is a function of the head fire rate and wind speed. The flank fire rate is the rate at which fire moved perpendicular to the wind. It is calculated using the head fire rate, back fire rate and an aspect ratio to form an ellipse. One method of calculating the aspect ratio for the ellipse is as a function of wind from experimental data.

The Rothermel model forecasts how ground surface fires will spread while the FBP can model both ground surface and tree crown fires. The FBP does not have as many parameters to describe the fuel as the Rothermel model. This means less data are required to run an FBP simulation. However, the FBP was originally designed to only work with fuel types present in Canada [50]. Land cover fuel properties for the Rothermel model were more readily available, therefore simulating wildfires in areas such as Ireland was straightforward. This is why it was used for wildfire modelling this thesis.

2.1.4 Other Approaches to Wildfire Modelling

There are many more approaches to wildfire modelling. Some of these include distinct methods such as QUIC-fire, the use of Kalman Filters and Deep Convolutional Neural Networks (CNNs).

QUIC-fire took the approach of modelling how much oxygen was available for combustion [51]. It did this using parameters such as the density of air and advanced wind physics. Kalman Filters have been used in conjunction with unmanned aerial vehicles (UAVs) to model wildfires [52]. This was done by sampling the temperature in multiple areas of a wildfire using UAVs. A Kalman Filter estimated propagation behaviour and rate of spread

using the temperature gradients from the samples [53]. Deep Convolutional Neural Networks (CNNs) have also been utilised to predict wildfire spread in California, USA [54]. The CNN learns propagation patterns of wildfires from training on historic datasets.

Section 2.1 described different wildfire models. The Rothermel model was selected to model wildfires in the following chapters as it was efficient, simplistic, had fuel data available for European land cover and was often cited in related literature. However, the FBP and other approaches are viable alternatives.

2.2 Frameworks for Modelling Wildfires

This section discusses frameworks to model wildfires. Frameworks typically have two types of propagation approaches: cellular automata and continuous. Cellular automata propagation is restricted to a grid and a strict set of rules, where each cell is in a finite number of possible states and the state of one cell can influence the state of its neighbouring cells [55]. Continuous propagation occurs over continuous space and therefore is not restricted to a grid or finite states. This section also briefly mentions other tools that may assist with wildfire modelling.

2.2.1Cellular Automata Approaches to Wildfire Modelling

Cell2Fire is a program that utilises cellular automata to simulate wildfire spread [56]. Cell2Fire can be given a fire spread model such as the Canadian Forest Fire Behavior Prediction System, a starting fire, and information about the environment to begin simulating fire spread. As Cell2Fire is a program built on cellular automata it restricts the spread of fire to a regularly shaped square grid using the Moore neighbourhood, where each cell has eight neighbours (corner neighbouring cells inclusive) [57]. Cell2Fire generates the common elliptical shape of a wildfire for each cell based on the head fire, flank fire and back fire rate of spread (Figure 2.6). Once any point on the ellipse touches the centroid of a neighbouring cell that cell will also begin generating its own ellipse to continue propagation. The uniform grid presents a trade-off between resolution and computation time. Low-resolution grids are fast but may be less precise. High-resolution grids are computationally

intensive but have potential to be more precise. Cell2Fire was not used later in this thesis as alternative frameworks had more readily available fuel data for land cover in Europe.

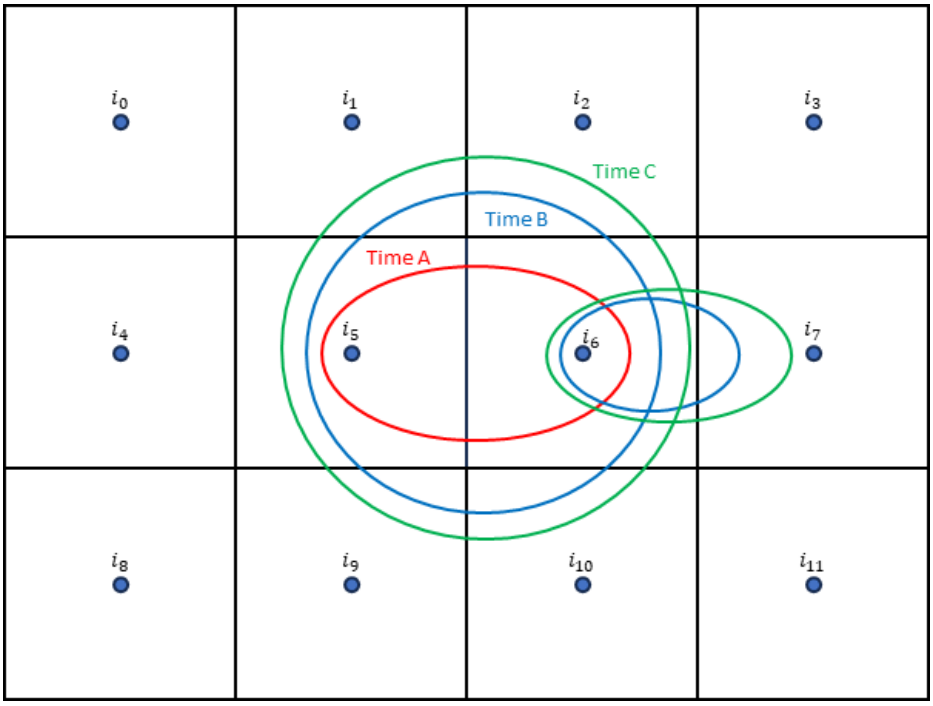

Figure 2.6: Example of how fire propagates from cell centroids in Cell2Fire. Each cell is given an index as a unique identifier. The fire propagates in an elliptical shape. The fire starts propagating from cell i_5 and reaches the cell centroid of i_6 at time A. The fire then propagates from cells i_5 and i_6 for times B and C. Image based on Cell2Fire: A Cell-Based Forest Fire Growth Model to Support Strategic Landscape Management Planning [56].

The Rothermel model has also been used in cellular automata frameworks [58]. In the referenced paper wildfires progressed from a cell's centroid to neighbouring cell centroids at a rate determined by the Rothermel model. Once a wildfire reached a neighbouring cell's centroid, that cell would also spread the wildfire to its neighbouring cells. Similar techniques were used in Chapter 3. A more traditional approach to modelling wildfires using cellular automata is described in Section 6.2.

2.2.2 Continuous Approaches to Wildfire Modelling

FARSITE is a framework for modelling wildfires over continuous space [59] [60]. Wildfires are represented by a polygon consisting of lines connecting vertices to draw its shape. A wildfire model is used to estimate the distance with the wind (head fire) that the wildfire will travel for each of these vertices over a set time step. An ellipse is used to represent the

area that the wildfire will spread to. The ellipse is generated for each vertex using the head fire's new position [61]. The eccentricity of an ellipse is increased with wind speed. An outer line called an envelope is then drawn tangential to the position of the head fire for each ellipse. This forms a new wildfire polygon at the next time step (Figure 2.7) [62]. This process continues for the required simulation duration. FARSITE finds the propagation rate of both ground surface and tree crown fires. The propagation rate of surface fires was calculated using the Rothermel model, while the propagation rate of tree crown fires was calculated using a model developed by Charles Van Wagner [63] [64].

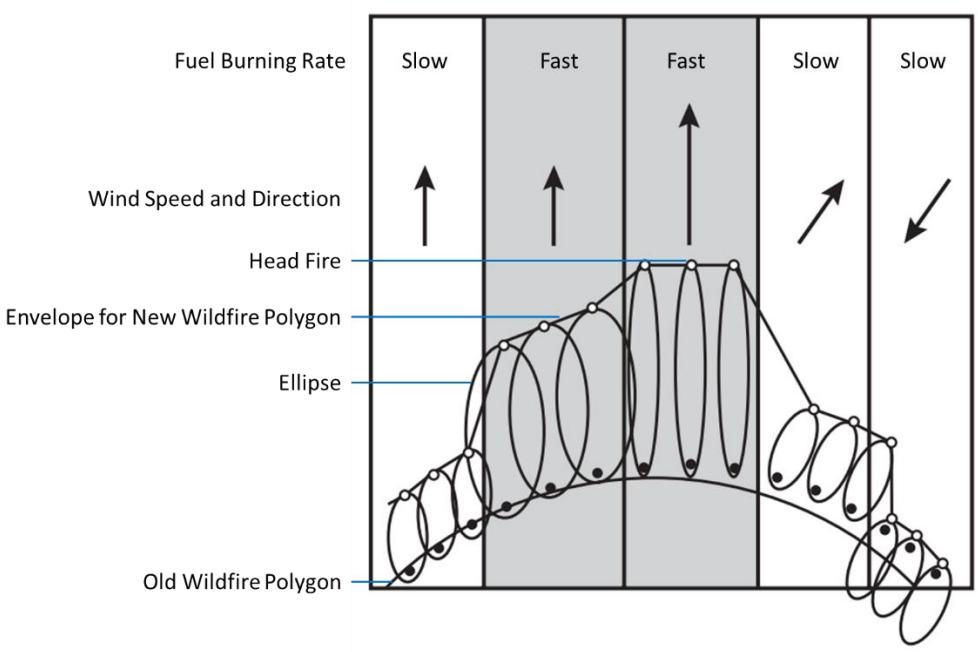

Figure 2.7: Ellipses used to propagate a wildfire from an initial polygon, at different wind speeds and types of fuel in FARSITE. The head fire is labelled. The envelope for the new wildfire polygon is also shown. Image based on FARSITE: Fire Area Simulator—Model Development and Evaluation [60].

Prometheus is another continuous framework for modelling wildfires [65]. Prometheus begins with an initial wildfire polygon. The Canadian Forest Fire Behavior Prediction System was used to calculate the propagation distance over a time step in the direction of a head fire for each vertex. Just like FARSITE, the propagation distances are used to generate an ellipse around each vertex. A fire line polygon is created using a tangential envelope of the set of ellipses. Overlapping edges are removed from the polygon. Prometheus was not selected for any comparisons in this thesis due to it using the Canadian Forest Fire Behavior Prediction System instead of the Rothermel model for propagation.

ForeFire simulates wildfire spread over a continuous space [66] [49]. ForeFire uses a fire model, such as the Rothermel model, to compute propagation. ForeFire running the Rothermel model was used for comparison in Chapter 3. It takes spatial environmental data over a selected region of terrain as an input. The environmental data contain wind data in the form of zonal (West-to-East) and meridian (South-to-North) wind speeds, elevation data regarding the terrain, and a numerical index representing different land cover types for the terrain. All data inputted into ForeFire are in the form of metric units. If the fire spread model required imperial or United States customary units (such as the Rothermel model), they need to be converted before calculating fire spread and converted back to metric units once rate of fire spread has been calculated. ForeFire is seeded with information about the fire's starting location, the spatial resolution, and the duration of the simulation. During the simulation ForeFire calculates the propagation rate of fire using the current fire location and environmental inputs. The fire perimeter is comprised of nodes called markers that ForeFire uses to continue spreading for the next iteration. Each marker has a propagation vector representing its movement direction and speed of propagation. The speed at which the marker moves at is determined by the fire spread model. If the distance between markers is greater than a user inputted spatial resolution, more markers will be created and redistributed along the fire front. If the distance between markers is less than a quarter of the spatial resolution, they will be merged into one marker (Figure 2.8). For this thesis the spatial resolution of ForeFire was set to the same as its default example resolution of $30m^2$. On completion, the software returns the final set of coordinates (known as markers) describing the location of the fire front. The use of markers instead of a grid allows for increased resolution as their position is on a continuous plane, limited by the resolution of the input data and the user inputted spatial resolution. However, the continual placing, removing, and moving of markers add extra computational burden on the simulation.

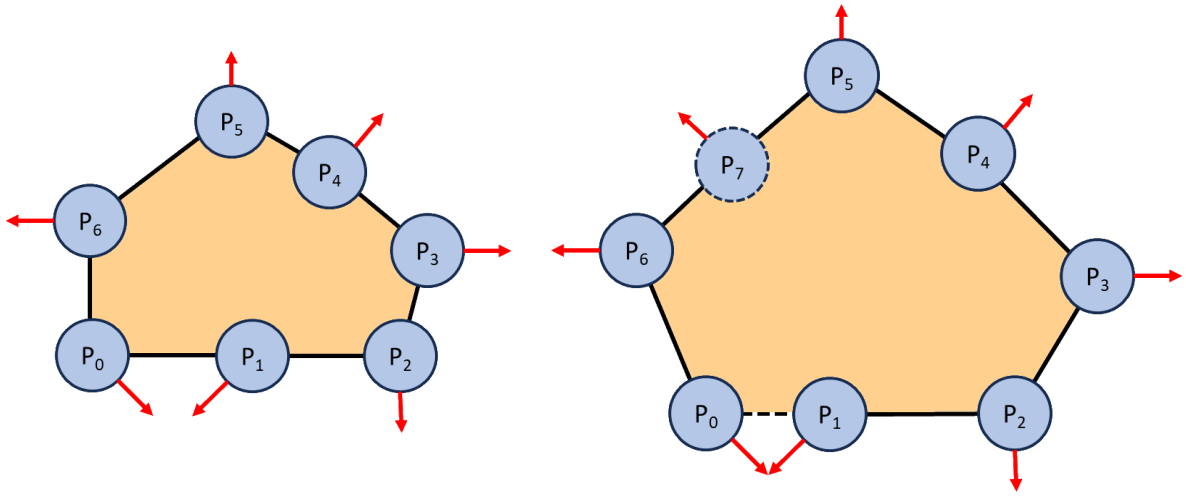

Figure 2.8: Example of ForeFire markers at time step 0 (left) and time step 1 (right). The fire line (black line), markers (blue circles), the direction of spread for markers (red arrows), and where the fire has been (orange) are all shown. In the example markers P0 and P1 will combine into one marker as the distance between them is a quarter of the spatial resolution. Marker P7 will be created as the distance between markers P5 and P6 is greater than the spatial resolution.

FARSITE and Prometheus are similar frameworks for simulating wildfires. The primary difference between the two is that FARSITE uses the Rothermel model and Prometheus uses The Canadian Forest Fire Behavior Prediction System. The Prometheus model was not used for any comparisons in this thesis as it used the Canadian Forest Fire Behavior Prediction System. Just like FARSITE, Cell2Fire and ForeFire could also use the Rothermel model. ForeFire was selected for comparison in Chapter 3 due to its readily available data containing fuel properties for Europe.

2.2.3 Additional Tools to Assist in Wildfire Modelling

There are many frameworks that can be used to model wildfires but there are also other tools that can help with seeding the model itself. 'Behave' is a fuel modelling software which can be used to test and adjust different fuel types [67]. It can run models based on preexisting fuel properties saved in the program or the user may also input their own data as fuel properties. This section was included as it might be useful for testing and creating new fuel descriptions.

2.3 Physical Fire Models

Wildfire modes can be developed using experimental data gathered from observing fire propagating on fuel beds. Fuel beds can be split into three size categories: large (greater than $100m^2$), medium (between $1m^2$ and $100m^2$) and small (less than $1m^2$). Small fuel beds are used to model fires in Chapter 5 of the thesis.

2.3.1 Large Sized Fuel Beds

A real wildfire typically burns on a large fuel bed. Wildfire models have been derived from real wildfires in the past [40]. The referenced study used satellite imagery to detect wildfires and measured the rate at which the wildfires propagated. The satellite imagery had a spatial resolution of $375m^2$ and a temporal resolution of 12 hours. The data were recorded and correlated with fuel types (land cover) and seasonal variation.

The use of a large fuel bed such as a real wildfire is straightforward and effective as the model does not need to extrapolate far from the data to the fire being simulated. These models typically define the fire in terms of rate of spread given a set of conditions such as fuel type. Data from these sources is limited and does not allow independent control of the physical parameters affecting the burn rate. It is impractical to manually change variables such as fuel, wind and slope in a large fuel bed, making observations regarding their effect on propagation rate mostly dependent on the environment.

2.3.2 Medium Sized Fuel Beds

The foundation of the Rothermel model was developed using a theoretical basis [68]. However, the remainder of the model was constructed based on experimental data [69]. Experimental data were collected in the Northern Forest Fire Laboratory, Missoula, Montana, USA. Medium sized fuel beds comprising of forest material such as pine needles, leaves and rotten wood were used where they had a length of either 2.44 or 3.66 metres, a width of 0.91 metres and a depth of 1.37 metres. Data from the ignited fuel beds were recorded. The data included the mass lost in the fuel bed due to ignition, progression of the

fire front, density of radiation on the surface of the fire, flame shape, air temperature, dewpoint temperature and air velocity [70].

Other studies explored the effects of slope and wind speed on fire propagation using pine needles and woody fuels as a fuel bed [71] [72]. Experiments have also been conducted on different fuels such as peat [73] [74]. The referenced research focused on the effect of moisture content on the propagation rate of the fire. This produced a smouldering fire where propagation rate of the fire, changes in temperature and the mass loss due to burning were recorded.

Variables in medium sized fuel beds such as fuel, slope and wind can be modified a lot easier than in large fuel beds. This allows changes in the rate of spread based on these variables to be observed. These fuel beds do have disadvantages of still being large and sometimes costly. They require an empirical or statistical model, so the observed behaviour can be scaled up to the size of a wildfire. Errors from collected data in medium sized fuel beds may get larger when scaling up to the size of a real wildfire.

2.3.3 Small Sized Fuel Beds

Small fuel beds consisting of paper have been used as a fuel for modelling fires. The work in Chapter 5 uses paper as a fuel bed. The following section reviews other work which took similar approaches.

Experiments were completed where multiple filter papers were burned in laboratory conditions at different orientations, to gather data on how slope affected burn rate [75]. In the referenced study the orientation and initial temperature of paper were controlled. The rate of spread, flame length and char length were recorded.

Papers soaked in an oxidising agent, potassium nitrate (KNO_3) were burned in a different experiment [76]. The resulting fire front closely followed self-affine scaling statistics which estimated the ratio of length to width of the fire. A similar study was completed where paper treated with KNO_3 was burned with no wind, on a flat surface (no slope) [77]. In that

study images were manually processed to record the expansion of the burn scar on the paper.

A stochastic grid-based fire spread model was developed in a similar experiment [78]. The referenced study used wax paper instead of KNO_3 treated paper as fuel. It focused on measuring burn rate as a function of slope. Rate of spread was measured by manually analysing recorded videos of the paper burning.

Small fuel beds have an advantage of being straightforward to configure and burning times are short. The disadvantage of small fuel beds is the requirement for a model, to scale up the results to the size of a real wildfire. Errors from collected data have potential to be greater than the medium fuel bed after scaling up to the size of a real wildfire.

2.4 Fire and Rescue Servies

Fire and rescue services across the globe have multiple responsibilities. These responsibilities typically include suppressing wildfires, removing ground debris that could fuel a wildfire, locating wildfire ignition points, determining what caused a wildfire and rescuing civilians and animals from dangerous environments including wildfires.

2.4.1 Wildfire Interventions Used in Ireland

In Ireland there are five main bodies that manage wildfires. They are the Irish Fire Service, Civil Defence, Irish Air Corps, Coillte and the National Parks and Wildlife Service.

The Irish Fire Service manage both small domestic house fires and wildfires [79]. Fire Officers develop strategies for managing fires using their expertise and information gathered from firefighters at the scene. They also perform risk analysis based on the environment and weather. They remove flammable ground debris from areas that are susceptible to wildfires. Firefighters are equipped with tools to traverse a burning environment and to help manage fires. They have three main vehicles at their disposal. These consist of fire engines containing large volumes of water, fire command support units

which are used as a central hub of information during wildfires and smaller vehicles which can be used to transport fire service members at speed. The Civil Defence work closely with the Irish Fire Service to provide additional support [80].

The Irish Air Corps are part of Óglaigh na hÉireann (the Irish Defence Forces) [81]. They usually assist the Irish Fire Service with wildfires. They are equipped with electronic tablet devices which allow them to communicate with the Irish Fire Service. Their primary vehicle for managing wildfires are Agusta Westland AW139 helicopters equipped with Bambi-Buckets [82] [83]. Using the helicopter, the Irish Air Corps can gain information regarding the fire from an aerial perspective which they can then relay to the Irish Fire Service by contacting a Fire Officer on the ground. The Bambi-Bucket can be used to collect water from nearby deep-water sources such as lakes or the ocean. The water is dropped from a height in front of an advancing flame to extinguish it or reduce its propagation rate.

Coillte is an Irish state-owned commercial forestry business in Ireland (they also maintain some forests for recreational activities) [84]. Coillte manages approximately 5% of land on the island of Ireland [85]. If a wildfire is propagating on this land, Coillte will assist the Irish Fire Service and the Irish Air Corps supressing it.

The National Parks and Wildfire Service is a service developed by the Irish Government Department of the Environment, Heritage and Local Government [86]. They manage Irish National Parks, engage in Ireland's conservation responsibilities and are interested in the ecological damage and subsequent recovery from wildfires. They oversee 1% of land in Ireland [87]. If a wildfire is burning on this land The National Parks and Wildlife Service will help the Irish Fire Service, and the Irish Air Corps manage it. Details regarding wildfire exercises which involved these institutes are discussed in Chapter 6.

There are three main ways that wildfires are fought in Ireland. The first of which is by applying water to the fire or the surrounding terrain [88]. This is done by the Irish Fire Service through a hose connected to a fire engine and the Irish Air Crops from a Bambi-Bucket attached to a helicopter. When water is added to an existing fire it may reduce the heat in that area slowing down combustion. If the water is applied to the surrounding fuel,

it will increase the amount of moisture in this fuel, which heat from the existing fire will need to evaporate, before it can propagate to this fuel. This may slow or completely stop combustion. Water may also briefly reduce the supply of oxygen to the fire, but this impact is minor.

Another approach to managing wildfires is creating fire breaks. Fire breaks are areas of terrain which do not support continual combustion [89]. Some natural fire breaks already exist on the terrain such as water features and roads while other fire breaks can be created on demand by the Irish Fire Service or Coillte by manually removing foliage. Fire breaks remove the fuel component from the fire tetrahedron required to sustain combustion.

The last method of managing fires in Ireland is through fire beating. This process involves a person manually hitting a wildfire with a tool called a flapper [90]. The Irish Fire Service and the National Parks and Wildlife Service usually perform fire-beating. It works by depriving the fire of oxygen. It may also cool the fuel and make the fuel slightly denser which both also slightly slow the propagation rate.

2.4.2 How Fire Investigators Find a Wildfire Ignition Point

Determining the location of wildfire ignition points and propagation durations is critical for a wide range of fire management and fire ecological applications. One of the many ways to reduce the number of destructive wildfires is by preventing ignition. A key part of preventing wildfire ignition is educating people to help avoid human-caused wildfires [91]. Wildfires can be caused by humans through power equipment, campfires and arson [92]. Human caused wildfires tend to occur in areas of high population and high road density [93].

The cause of a wildfire is normally determined by a fire investigator searching the scene for debris that may indicate how the fire started [94]. To find debris a fire investigator must locate the wildfire ignition points. Currently wildfire ignition points are found manually by fire investigators using several fire pattern indicators. The angle of char, vegetation curling, and scar shapes are all examples of fire pattern indicators that fire investigators use [94]. The ignition points of wildfires could be used as a risk assessment tool to predict where

wildfires might appear in the future. This can influence fire management decisions to place fire breaks or clear forest litter in areas of high risk ahead of time.

Char forms when a fire burns standing fuel such as a tree. The angle of charred fuel when the fire is burning downhill or against the wind, should be parallel to the ground across the standing fuel. If the fire is burning uphill or with the wind the standing fuel will be charred at a greater height in the direction the fire is moving in, than the direction it is coming from, creating a sloped charring pattern at a greater angle than parallel to the ground.

Vegetation curling occurs when oncoming fire heats up vegetation. This is due to vegetation drying out and shrinking when it is exposed to the heat, causing the vegetation to curl towards the direction of the fire.

During the early stages of wildfire progression in windy conditions, wildfires tend to form burn scars in a 'u' and 'v' shape (Figure 2.9). Burn scars with a 'u' or 'v' shape pattern tend to have an ignition point at the base of the 'u' or 'v'. These are just a handful of fire pattern indicators used to find the ignition point of a wildfire but there are many more that fire investigators have at their disposal.

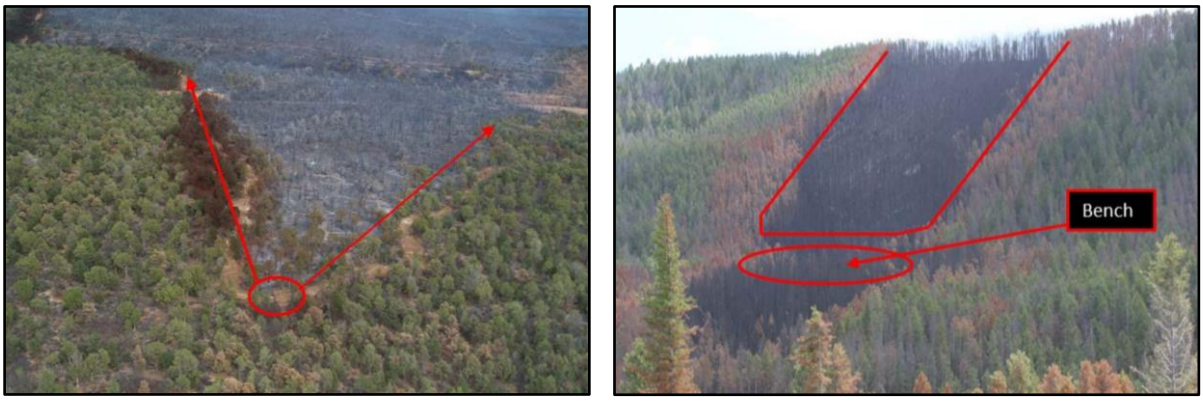

Figure 2.9: Example of a 'v' shaped burn pattern (left) and 'u' shaped burn pattern (right). Images from Guide to Wildland Fire Origin and Cause Determination [94].

Chapter 4 presents an automated method of detecting wildfire ignition points which could assist members of the fire service.

2.5 Geographic Information Systems

Geographic Information Systems (GIS) are a group of hardware and software that are used to record, analyse, modify and visualize geographic data [95]. There are two main sources of GIS data which are satellite imagery and geographic features files. GIS data tends to be geocoded which means that values (such as pixels) in the data have an associated longitude and latitude coordinates. The data can also be projected to other coordinate systems. In this thesis the EPSG:2157 Irish Transverse Mercator projection is used for any GIS related computing [96]. A coordinate in the EPSG:2157 projection has two parts, the first value represents the West-to-East position in Ireland at a resolution of 1 metre (also known as an Easting), and the second value represents the South-to-North position in Ireland at a resolution of 1 metre (also known as a Northing). GIS allows wildfire modelling frameworks to be linked to the real world.

2.5.1 Satellite Imagery as Data Sources

Satellites are a source of image data that can record information regarding Earth's surface. The European Space Agency is one source of satellite imagery. The images taken by their fleet satellites [97]. In this thesis all satellite data were recorded by the Sentinel-2 group of satellites [98]. The Sentinel-2 group of satellites consist of three separate satellites (as of thesis publication), which all record the same type of data. Multiple satellites are used to get semi-frequent images of Earth's terrain as non-geostationary satellites do not remain still over a certain section of Earth but instead orbit it. Saved Sentinel-2 satellite data contains 12 bands which all detect different wavelengths on the electromagnetic spectrum, including visible light. The bands have a varying resolution from $10m^2$ to $60m^2$ of Earth's surface. A subset of the bands can be combined to compose coloured images.

Satellite data are commonly used as an input for wildfire models. Geocoded satellite data can be aligned with other existing geocoded data at the same coordinates including digital elevation maps [99]. A tool called SNAP (Sentinel Application Platform) Desktop provides a method of doing this [100]. Machine learning models trained on other forms of data can also be used to generate additional geocoded data. Models trained on the CORINE

(Coordination of Information on the Environment) Land Cover classes have been used to predict land cover types using the bands from satellite images (A.1 Generating a Land Cover Map using Satellite Data (Credit: Gourav Misra)) [101]. A typical GIS workflow consists of classifying areas in satellite imagery to land cover types and then converting this to a set of fire fuel properties. Irish wind data from primary sources tend not to be geocoded but instead are a fixed value retrieved from the Irish Meteorological Service called Met Éireann [102]. Satellite data are used in Chapter 3 as an input to the Rothermel model.

2.5.2 GIS Features

GIS is also used to describe features and their positions. There are three common types of features described in GIS, which are points, lines and polygons. Points are zero-dimensional objects in space. An example of this would be a wildfire ignition point. Lines are one-dimensional objects in space, such as an active fire line. Polygons are two-dimensional objects in space. A wildfire burn scar would be an example of this.

The features described above are usually stored in a file format such as a Shapefile, GeoJSON (Geo JavaScript Object Notation), or KML (Keyhole Markup Language). They are geographically represented by one or more coordinates. These features may also have additional metadata saved with the feature which may include information about it. The listed file formats were used to display fire lines in Chapter 6.

A list of burn scars throughout Europe has been compiled [103]. They were used for comparison in Chapter 4.

2.6 Computer Vision

Computer vision is a technique used to process, analyse, extract, and interpret information from an image [104] [105] [106] [107]. Early research in the field of computer vision began with digital image processing [108]. In 2000 an open-source computer vision software called OpenCV was developed [109]. It is widely used in computer vision. Machine learning and traditional methods are the two main approaches to computer vision.

Computer vision has been used to perform various tasks such as classifying images of hand gestures from Irish sign language to their associated letters of the alphabet [110]. It has also been used to detect and count cars moving on roads [111]. In Chapter 5 of this thesis a standard computer vision pipeline was developed to automate tracking of a fire line on burning paper.

2.6.1 Machine Learning applied to Computer Vision

Advancements in machine learning (ML) during the 2010's sparked the use of machine learning to perform object detection tasks with architectures such as YOLO (You Only Look Once) [112]. Platforms like YOLO were used to perform object detection on animals such as insects; even using edge computing [113]. This may have application in monitoring wildlife recovery after a wildfire. These platforms train on a dataset of labelled images. Typically, a Convolutional Neural Network is modified over many training epochs to classify and sometimes locate objects in images that were not in the training dataset [114].

The detection of fire and smoke using machine learning has been tested. In [115] a modified version of the YOLOv5 platform was developed called YOLOFM where it detected both fire and smoke. A separate team in DecaMap developed a YOLO model to help firefighters detect humans, animals and cars from live camera feeds provided by drones scanning the environment during firefighter training exercises [43]. This is mentioned in Chapter 6.

2.6.2 Traditional Computer Vision

Traditional computer vision consists of a pipeline containing the steps: capture, enhance, segment and classify/measure [113]. The capture step optimises the physical conditions to capture the highest quality image possible (e.g. camera sensor, lens, lighting). The enhance step involves various image processing techniques which makes the following steps easier to perform (e.g. contrast enhancement and thresholding). The segment step involves characterising areas of interest in the image; changes made during the enhancement stage

should help with this. The parameters measured for each area of interest can then be used to classify images into one of multiple predefined categories. Measuring involves using the segmented area of interest to record data that can be used in further analysis. Properties of traditional computer vision pipelines are typically manually designed and adjusted. This gives greater control over the entire process compared to machine learning techniques.

It is possible to detect fires using traditional computer vision. In [116] 16 videos were downloaded where some contained fires and others did not. The presented algorithm attempted to detect the location of any fires in a particular frame. Fires were detected based on their visual properties. The first visual property used was a fire's tendency to flicker at a rate between 1Hz and 10Hz. Next were the colour values in forms of RGB (Red, Green, and Blue) along with HSV (Hue, Saturation and Value). Changes in the colour of pixels was also used to detect wildfires along with other steps in the algorithm. From the 16 videos tested in this paper it correctly predicted 15 of the results with it incorrectly predicting a fire in one video where there was none. Similar properties were used to detect fire using traditional computer vision in Chapter 5.

Smoke is a key part of most burning fires that can also be visually detected. Single frames from 15 different videos were analysed to detect if smoke was present [117]. Multiple visual properties regarding smoke were used to detect it. The first of which was the colour and brightness. The behaviour of smoke was also considered, such as smoke tending to rise upwards and diffuse into the air. There were also additional steps used in the algorithm. From the 15 videos it correctly predicted if smoke was present in the selected frame for 11 of the videos. Similar techniques for smoke detection using traditional computer vision were also used in Chapter 5.

Tracking how a wildfire propagates, after wildfire detection is an important part of wildfire monitoring as it provides information on its propagation behaviour [118]. Human occupied watch towers, sensors such as thermometers, and Kalman filters can provide a method of tracking wildfires without using computer vison [52]. However, computer vision techniques such as optical flow can also be used to track wildfires. An optical flow algorithm was designed to detect fires and distinguish between the fire and other distractions using the

properties of fire in [119]. In Chapter 5 fire propagation is tracked using traditional computer vision techniques such as background subtraction to detect changes in video frames which are caused by the fire.

2.7 Inverse Methods

An inverse method is a process that reverses an action to find the causes from the results [120]. Inverse methods can be applied to wildfires to find ignition points and propagation durations of a wildfire, terrain and weather data or even to construct a set of equations to model the wildfire. A list of different inverse methods is described in this section. In Chapter 4 inverse methods are used to estimate the location of ignition points and propagation durations of wildfires from their burn scars.

2.7.1 Approaches to Solving Inverse Problems

Many approaches can be taken to solve inverse problems. These include brute force search, Monte Carlo methods, time reversal algorithms and genetic algorithms.

Brute force search checks all possible solutions of a problem in an orderly fashion to find the optimal one [121]. While checking all possible solutions the current best performing solution is saved. Any solution that performs better than the existing best performing solution replace it. Given an infinite amount of time brute force search will find the optimal solution. Typically, brute force search is not practical in problems that are very high-dimensional as they may have many possible solutions.

Monte Carlo methods can also be applied to solving inverse problems [122]. Monte Carlo methods try random solutions, while saving the best performing solution. Given an infinite amount of time Monte Carlo methods will also find the optimal solution, but it is possible that this may take longer than brute force search as Monte Carlo may test the same solution more than once. It can however arrive at local optimum or the real optimum in a shorter time.

Time reversal algorithms can be used to solve inverse problems when the method used to produce the output is known and invertible [123]. This process begins using the outputs and runs through the algorithm that produced them in reverse order, undoing changes that were made at each step in the algorithm. Time reversal algorithms may not produce accurate results if there are any errors (sensitive to noise) in the output data or the algorithm is lossy (loses precision).

2.7.2 Genetic Algorithms

Genetic Algorithms (GA) provide an efficient means of searching a large space for potential solutions to a problem, which includes inverse problems [124]. They are inspired by natural selection, which is the process commonly theorised to drive biological evolution [125]. GAs can be used to find a good solution to an optimisation problem that may otherwise take too long to compute using brute force search [126]. Even though a GA may produce a good solution, this does not necessarily mean that it will always produce the optimal solution for the problem every time within a set number of generations. GAs start with an initial population of what are called chromosomes. In GAs a chromosome is a possible solution to the problem the GA is trying to solve. Each of these chromosomes consists of attributes that are called genes. The genes in GAs encode the characteristics of a chromosome. A gene is a set attribute of the solution (chromosome) to which it belongs. Before running a GA, a maximum number of generations is usually set to ensure it halts, typically the more generations a GA is run the higher chance it has of finding better solutions. A set of chromosomes are created (usually randomly) when starting a GA in a process called initialisation. Chromosomes in the population are then ranked using a fitness function for that GA. The chromosomes with higher fitness scores are typically selected through a process called selection. These selected chromosomes go through a process called crossover where they swap genes with other selected chromosomes. This creates a new set of chromosomes that could potentially offer better solutions. A process called mutation then usually occurs where some genes are changed at random, providing diversity in the new set of chromosomes. The new set of chromosomes form the population of the next generation, and the process repeats to improve the set of solutions iteratively. A similar

approach to this general example of a genetic algorithm was used in Chapter 4 to estimate the most likely ignition points and propagation durations for burn scars.

GAs have been applied to wildfire modelling before in different ways. A genetic algorithm has been used to fit fuel and wind values, which are inputs for the Rothermel model [127] [46]. In the referenced paper, a population of chromosomes that contained a set of input values required to run the Rothermel model was generated, with each input value acting as a gene. Every chromosome in this population was simulated using the FARSITE simulator running the Rothermel model with every gene in that chromosome containing different fuel and wind properties [60]. The outputted wildfire of this simulation was compared to the real wildfire using the fitness function, which calculated the area of overlap between the simulated and real wildfire to get the fitness score. Chromosomes with a high fitness score were used in crossover to generate new chromosomes. The input values also had a chance to mutate. The best chromosomes were saved and preserved during each generation in a process known as elitism. This process repeated for multiple generations to optimise the input values of fuel and wind for the Rothermel model so it would produce a burn scar like the real one. This referenced literature claims without GA calibration the mean prediction error between the rate of spread of the actual fire and the simulated fire was 95%. While with GA calibration, the mean prediction error between the rate of spread of the actual fire and the simulated fire was 6.03%.

Genetic algorithms have also been used in a designed scenario to estimate the optimal order for one set of firefighters to combat 10 wildfires in Braga, Portugal [128]. The chromosomes consisted of ordered genes with each gene representing a wildfire; the order of these genes is the order in which each of the wildfires should be combatted. To begin, an initial population of chromosomes was created. A fitness score was computed for each of these chromosomes. A set of chromosomes were chosen by fitness score using tournament selection. Ordered crossover was then performed on the chosen chromosomes ensuring that the order of the genes remain intact, followed by inverse mutation. This created a new generation of chromosomes, and fitness scores were then computed. The described process continued for a preselected number of generations. This outputted an optimised order for firefighters to manage wildfires based on the travel distance between wildfires and

the properties of the different wildfires. The optimal output in this paper claims it could have saved 160.294km^2 of the forested area from burning out of 161.248km km^2 which is 99.41%.

The application of genetic algorithms to estimate the location of wildfire ignition points and propagation durations simultaneously is a novel study described in Chapter 4.

2.7.3 Inverse Approaches Applied to Wildfire Modelling

Inverse approaches have been applied to wildfire modelling in different ways. Typically, this uses historical data to inform future predictions about wildfire propagation. An example of this is an algorithm to find the burnt area of a wildfire given its active fire line, which can inform the direction the fire is spreading [129]. There have been attempts to use satellite remote sensing to identify the sources of wildfires, but these have considerable limitations.

A conditional Wasserstein generative adversarial network (cWGAN) was iteratively trained on WRF–SFIRE (Weather Research and Forecasting-Wildfire Spread Simulation) to provide a forward spread model [130]. It was used to estimate the fire arrival time at a spatial location using satellite data [131] [132]. As part of the forward prediction the cWGAN estimates the ignition times and locations of a wildfire. The cWGAN was tested on wildfires that had a singular ignition point. The main aim of this work was to improve the forward prediction of the model rather than identify the point and time of ignition.

Fire line progression can be modelled as a set of potential active fire line perimeters that intersect detection points from satellite imagery [133] [134]. A likelihood map can be generated and from this, the location of the smallest perimeter and time of ignition can be estimated. The method was tested on wildfires with a single ignition point. The main goal of this research was to generate artificial fire history to inform a simulation starting from an active fire line. A similar technique was used to model the introduction position and date of strains of watermelon mosaic virus (genus *Potyvirus*, family *Potyviridae*) located in Southeastern Frace [135].

Satellite images have also been used in the past to improve recorded data on wildfire ignition points and the propagation duration of that wildfire [136]. Satellite images of the wildfires were overlaid on the recorded wildfire perimeters stored in existing databases. The area of the wildfire in the earliest satellite image corresponding to that wildfire in the database was recorded as the wildfire's ignition point. The start and end dates of the wildfire were dated as the earliest and latest satellite images containing the wildfire. This can provide accurate data in terms of the ignition point location and propagation duration. However, it requires cloud free satellite data of a wildfire at the beginning and end of a wildfire to perfectly predict the wildfire ignition points and propagation duration. This is not often possible as non-geostationary satellites do not remain over the same area of terrain constantly, and there are not enough satellites to have a constant feed of cloud free satellite imagery. However, there are alternate methods to detect wildfires in areas of cloud cover such as the use of machine learning algorithms to classify smoke [137].

A dataset of environmental features between June 2012 to September 2012 were categorised based on the presence or absence of fire [138]. Long Short-Term Memory and Gated Recurrent Unit models trained on the data were then used to estimate likelihood of wildfire ignition occurring given environmental features. This technique could be applied to region of interest showing the likelihood of fire. The main aim of this paper was to estimate when a wildfire ignition may occur.

Inverse models were also used to estimate the date and point of introduction of a phytopathogenic bacterium (*Xylella fastidiosa*) [139]. Spread of the phytopathogenic bacterium was simulated on a grid. The algorithm initially assumed each cell had an equal probability of being the point of introduction. Propagation of the phytopathogenic bacterium was simulated from these points of introduction. The difference between observed spread of phytopathogenic bacterium and the simulated spread was found. This information was used to adjust the points and date of introduction. This process is repeated iteratively and converges on a point of introduction. Single points of introduction were tested in this study.

From a literature review no papers have described an automated method with the goal of estimating wildfire ignition points and the propagation durations using wildfire burn scars. Discussions with local fire officers also confirmed this and revealed that an automated method for determining wildfire propagation durations, would be a useful tool for them. After a wildfire has finished burning, its burn scar remains on the terrain until flora regrows, which tends to occur over a long period of time [140]. Imagery of wildfires from satellites change rapidly as the wildfire propagates on the terrain. Therefore, even burn scar data recorded a long time after a wildfire can still be analysed for the purpose of estimating the ignition points. In Chapter 4 satellite data from Copernicus are used for obtaining terrain properties and to compile a dataset of real historic burn scars [103]. However, satellite data are not fundamental to the method, which would work using terrestrial or drone mapping techniques. The technique presented in Chapter 4 is also not temporally limited to the frequency of satellite imagery.

2.8 Concluding Remarks

This literature review informed choices made in the remaining chapters. The Rothermel model discussed in Section 2.1.2 was used in Chapter 3 and Chapter 4 to model wildfires as custom fuels such as those found in Europe were compatible. ForeFire, discussed in Section 2.2.2 was used as a comparable framework for modelling wildfires in Chapter 3, as it contained data on fuel properties for land cover in Europe. In Chapter 5 experimental burns were performed. Filter paper was used as fuel; it is a small fuel bed and had the advantages discussed in Section 2.3.3 of short burn times and being cheap to construct and run. Wildfire interventions performed by professionals, discussed in Section 2.4.1 were implemented into a framework for modelling wildfires, described in Chapter 6. Chapter 4 discusses an automated method of estimating the location of wildfire ignition points which is different to the manual techniques described in Section 2.4.2. Satellite data were selected as input for the wildfire modelling frameworks in Chapter 3 and Chapter 4 as geocoded information about land cover and elevation can be extracted, as discussed in Section 2.5.1. A list of existing real European wildfire burn scars were analysed in Chapter 4. The wildfires burn scars were in the format of polygonal GIS features which were discussed in Section 2.5.2. Traditional computer vision, summarised in Section 2.6.2 was used to track

fire in Chapter 5 as it was easy to manually modify specific properties. Genetic Algorithms were chosen to perform the inverse problem of estimating wildfire ignition points and propagation durations given a wildfire burn scar in Chapter 4. This is because genetic algorithms can search a large space for potential solutions to a problem in a reasonable amount of time compared to other techniques as discussed in Section 2.7.2.

Chapter 3 An Efficient Method to Simulate Wildfire Propagation Using Irregular Grids

In this chapter a novel software platform called the Irregular Grid Software (IGS) was developed which allows for the simulation of wildfires on a configurable grid using physics based computational models for fire propagation. The aim of the IGS was to explore computational differences between different grid types and make a comparison with preexisting software. The goal was to improve computational efficiency while producing outputs like preexisting software, which would make future computationally intensive analysis quicker. The configurable grid was built using a Voronoi diagram. This approach provided flexibility in the grid description. The fire can spread between polygons, propagating throughout the grid. The configurable grid allows for cross comparison of both regular grids such as square, hexagonal and triangular, and irregular grids such as a randomly seeded Voronoi diagram and a newly developed focused grid. The focused grid was adapted to have greater detail (using smaller, more tightly packed polygons) in areas of higher importance which provides greater precision at the cost of extra computing time. The focused grid and ForeFire, an existing industry standard program were compared. The comparison included simulations of wildfires located in the Wicklow Mountains, Ireland, a region used by fire services for exercises. The performance of the grid-based techniques was examined using a set of experiments to characterise the model's response to key factors such as wind, elevation, and fuel type. The objective of this research was to compare the various grid types on the metrics of similarity with ForeFire and computational time, while also comparing the focused grid to ForeFire using the same metrics with multiple sample wildfires. From the sample wildfires tested in this chapter the results show that the IGS runs on average 34 times more quickly than ForeFire on the same computing platform while retaining an average result similarity of 80% with ForeFire. The IGS was purposely designed to use the same input parameterisation as ForeFire to allow for a fair comparison between both platforms.

There is a description of the resources used in Section 3.1. Section 3.2 shows how the IGS wildfire modelling framework works. A new method of checking the similarity of polygons

for comparing the IGS to ForeFire is discussed in Section 3.3. The different grid types are compared in Section 3.4. The focused grid is compared to ForeFire in Section 3.5. Section 3.6 contains a discussion regarding these results. This chapter's conclusions are in Section 3.7.

3.1 Resources

The Rothermel model was used by both the IGS and ForeFire to simulate wildfires. Therefore, IGS and ForeFire both required input data describing the land cover, land cover fuel data, elevation and weather conditions.

A Sentinel-2 satellite image of the Wicklow Mountains, County Wicklow, Ireland, was inputted into a random forest machine learning model to produce a land cover map (A.1 Generating a Land Cover Map using Satellite Data (Credit: Gourav Misra)). The land cover map was rectangular in an EPSG:2157 projection with its most Northwestern coordinate at EPSG:2157 (698998.1, 722771.812) and most Southeastern coordinate at EPSG:2157 (722773.635, 697886.5) which covered a total area of approximately 592km^2. Each pixel on the land cover map referenced an associated land cover type including pastures, sparsely vegetated areas, mixed forests, moors and heathland, urban fabric, waterbodies, clouds, and cloud shadows (Figure 3.1).

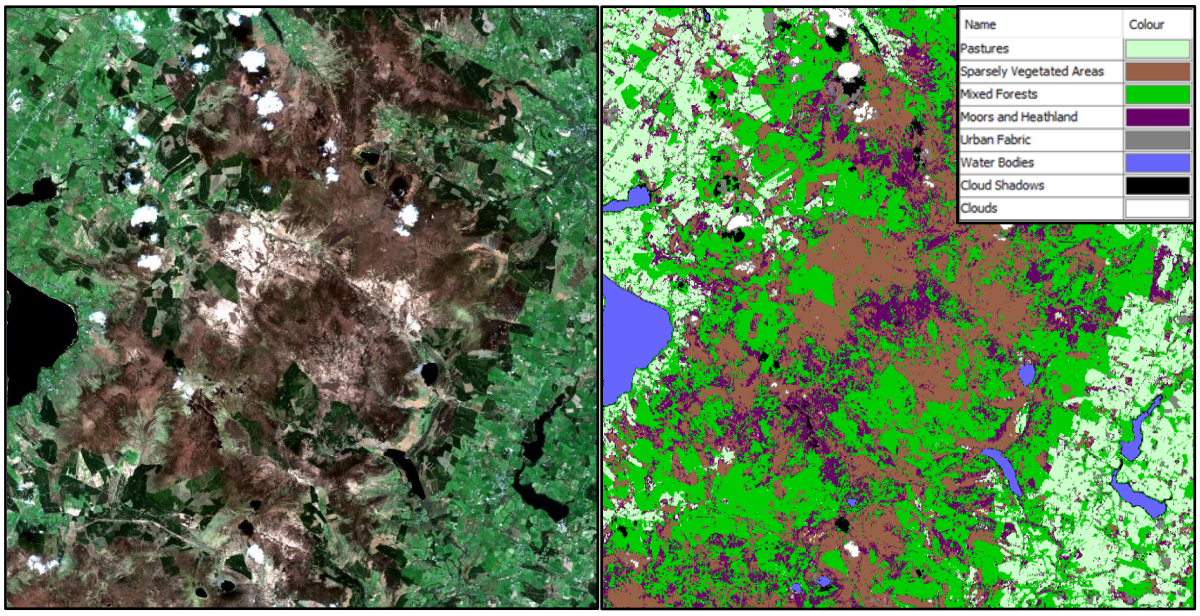

Figure 3.1: On the left is a Sentinel-2-based true-colour image of the study area, and on the right is a predicted land cover map for April 2022 (Copernicus 2022).

Each type of land cover was mapped to a set of physical attributes that were contained in a file with a lookup table used by both programs. ForeFire contains one of these files, called a "fuels file" [141]. The same file was also used by the IGS. The "fuels file" was indexed based on CORINE Land Cover classes [101]. The "fuels file" contained values for fuel particle density (kg/m^3), the ratio of fuel particle moisture content, fuel particle surface area to volume ratio (m^{-1}), fuel height (m), the oven-dry fuel load (kg/m^2), and fuel particle low heat content (J/kg) for the different land cover types. The Rothermel model used by the IGS and ForeFire had no mineral damping, and the moisture of extinction was set to 0.3 for all fuels in this chapter. These listed fuel properties were required to run the Rothermel model. The land cover map had two additional fuel types for when it was not possible to identify the land type due to clouds or shadows. These fuel types were not present in the simulation area, as it was cloud free, so they did not affect either program.

The Sentinel-2 satellite data and a geographic information software; SNAP Desktop's elevation band generator, were used to produce an elevation map of the area (Figure 3.2) [97] [100]. The elevation map gives the height in metres of elevation above sea level for each pixel in the satellite data.

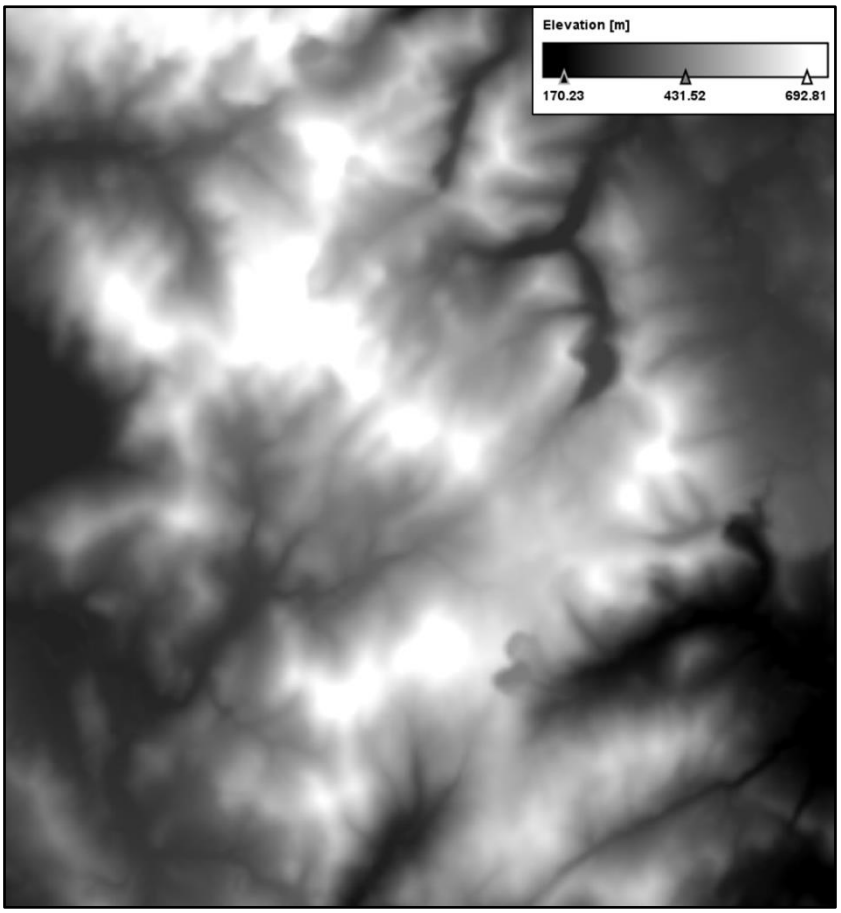

Figure 3.2: Elevation map of the study area where darker colours represent areas of lower altitudes and lighter colours represent areas of higher altitude.

For experimental simulation, a set of different wind speeds and directions were selected to allow for a comparison between ForeFire and the IGS under the same conditions. However, an API developed by DecaMap also allows for software such as Windy to provide live, forecasted, and historical wind data for simulations [142]. Wind data were then split into zonal (West-to-East) and meridian (South-to-North) wind speeds which are mapped at each pixel across the satellite data.

ForeFire contained a tool that allowed for the land cover, elevation, wind data, and metadata to be combined into a single NetCDF file [143]. NetCDF files are used to store array-oriented scientific data, typically geospatial data. The same NetCDF file was then used by both ForeFire and the IGS.

3.2 Simulating Wildfire Propagation Using the IGS

The IGS is a Python program with the aim of generating a grid-based fire spread model that could be compared to ForeFire. The use of a static grid allows for a model to compute fire spread without having to continually move and add markers during the simulation as found in ForeFire.

A Voronoi diagram was used to create an irregular grid. They have been used to simulate the geographic spread of disease in the past [144]. A Voronoi diagram takes in a set of points called sites. From these sites, edges (lines) are generated, located equidistant between each pair of sites, perpendicular to the direction between both sites. For the length of a particular edge the two sites that are on either side of it are the two sites that are closest to it; therefore, once the distance from any part of the edge is closer to a third different site, that edge stops. This creates contiguous polygons where every point within a polygon's perimeter is closer to that polygon's site than any other polygon's site (Figure 3.3). This means that each edge separating polygons is equidistant to both polygons' sites. The Voronoi diagram was generated using the efficient Fortune's algorithm implemented in the Foronoi (not spelled Voronoi) Python library [145] [146]. One of the biggest advantages of using a Voronoi diagram is the ability to create irregular simple contiguous shapes. This provides more flexibility in the grid structure where areas of importance can be more detailed.

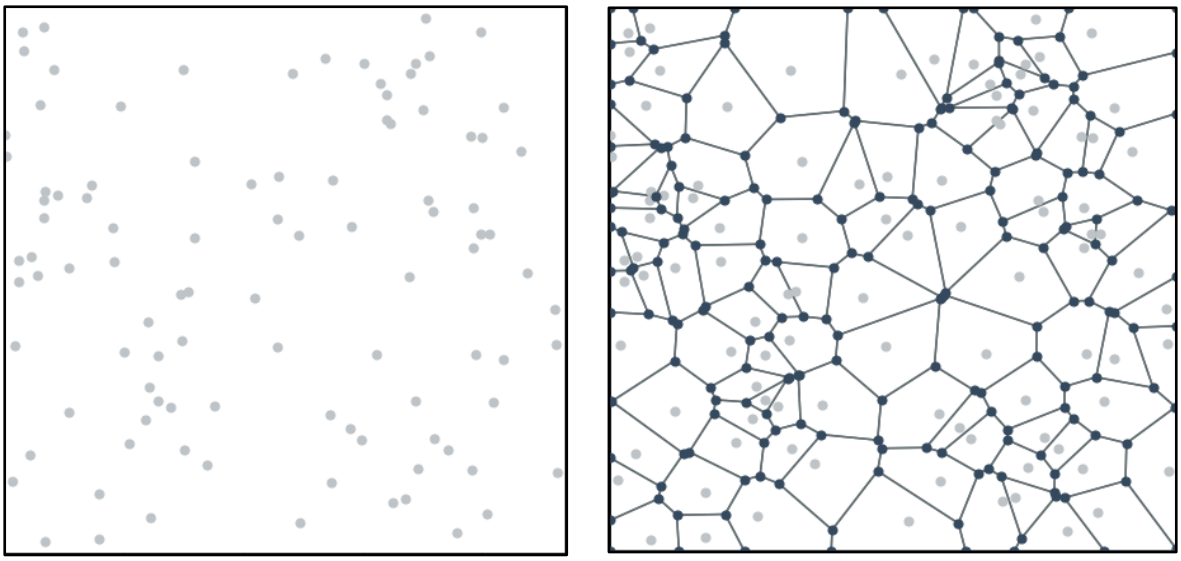

Figure 3.3: Positions of randomly placed sites (left) and the resulting generated Voronoi diagram of polygons (right).

The NetCDF file containing input data, created using ForeFire, is imported into the IGS using the Snappy Python library [147]. Snappy allowed for the NetCDF data to be read into Python lists used for computation. The generated Voronoi diagram was overlaid on top of the NetCDF data. The elevation, wind, and fuel values (per fuel index) were recorded for every pixel in all polygons of the Voronoi diagram. This was done using the scan-line polygon fill algorithm to extract environmental data from the pixels contained within the polygon [148]. The mean elevation, wind and fuel values of each polygon were then saved.

The IGS starts with a fire located at the site of a Voronoi polygon. The propagation of this fire towards the site of each bounding polygon is modelled using the Rothermel model. The Rothermel model calculates the rate of spread for the head fire (fastest moving part of the fire). Rates of spread for the flank fires (perpendicular to the head fire) and back fire (opposite the head fire) are not calculated. When the fire reaches the site of a bounding polygon, the fire then progresses towards the new neighbouring polygons from that bounding polygon. Fires may propagate towards each other simultaneously. Fire progress is recorded as a ratio of how far it has propagated towards neighbouring polygon sites and the total distance between the two polygon sites. This is known as the propagation ratio. When the Rothermel model is run, the propagation ratio from the ignited polygon to its neighbours will increase if the rate of spread is positive (Figure 3.4) (Eq. 3.1). When the propagation ratio is 1, fire will

reach the neighbouring polygon's site, and it will also begin spreading fire. The propagation ratio is calculated as follows:

$$(P_{ij})_{t+dT} = \frac{\left((P_{ij})_t d_{ij}\right) + (R_{ij} dT)}{d_{ij}},$$

(Eq. 3.1)

where i is an ignited polygon, j is a polygon that neighbours the ignited polygon, t is the current time (s), $dT > 0$ is the time step (s), $0 \leq (P_{ij})_{t+dT} \leq 1$ is the propagation ratio from polygon i to j at time $t + dT$, $0 \leq (P_{ij})_t \leq 1$ is the propagation ratio from polygon i to j at time t, $R_{ij} \geq 0$ is the rate of spread from polygon i to j calculated by the Rothermel model (m/s) and $d_{ij} > 0$ is the distance between the sites of polygons i and j (m).

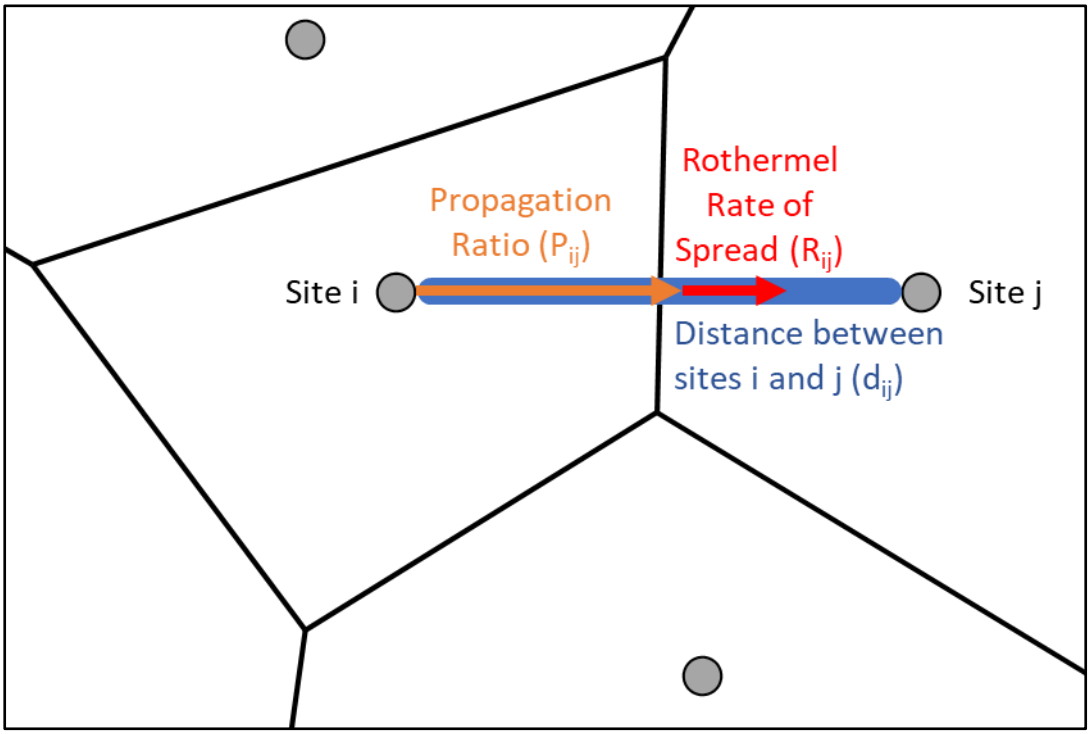

Figure 3.4: Example of the fire spreading from polygon i to j, where the blue line represents the total distance (d_{ij}) between sites i and j (m), the orange arrow represents the propagation ratio (P_{ij}) of how far fire has spread from site i to j, and the red arrow represents the fire's rate of spread determined by the Rothermel model (R_{ij}) from site i to j (m/s).

Every polygon has fuel data associated with it. The IGS uses the fuel data associated with the polygon the fire is spreading to during simulation. If the fuel data associated with the polygon the fire is spreading from were used, then fire would spread from that polygon to all its neighbours at an equal rate in conditions where there is no slope or wind. Another option

would be to get a weighted fraction between the two polygons. This method would increase computation time for simulating a wildfire. Therefore, the fuel properties from the polygon the fire is spreading to were used.

The Rothermel model requires data describing the fuel, the terrain elevation, and wind. The slope is found by getting the difference in elevation between the neighbouring polygon and ignited polygon, which is then divided by the horizontal distance between the polygon sites (Figure 3.5):

$$S_{ij} = \frac{H_j - H_i}{g_{ij}},$$

(Eq. 3.2)

where i is the ignited polygon, j is the neighbouring polygon, S_{ij} is the slope from i to j, H_j is the mean elevation of polygon j (m), H_i is the mean elevation of polygon i (m) and $g_{ij} > 0$ is the horizontal distance between the polygon sites i and j (m).

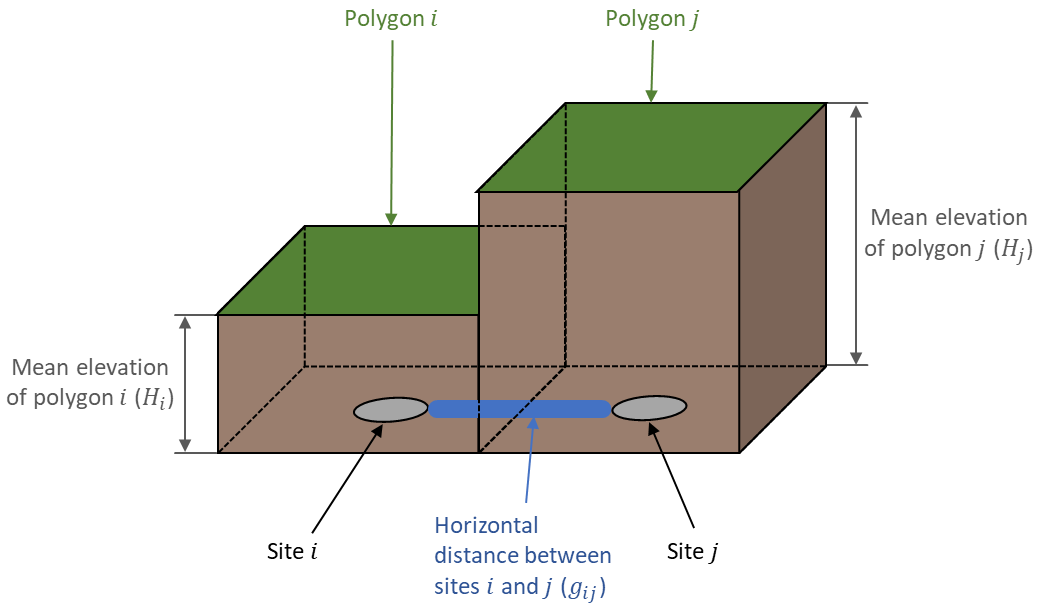

Figure 3.5: Variables used in (Eq. 3.2) to determine the slope from polygon i to polygon j.

The normal wind is the wind velocity at midflame height travelling in the same direction as the wildfire, if this value is negative the wind is travelling against the fire. This value is used in the Rothermel model. It is found by getting the dot product between the wind vector and

the unit vector describing the direction from the ignited polygon to the neighbouring polygon site's coordinates (Figure 3.6):

$$(W_{ij})_n = \begin{bmatrix} (\boldsymbol{W_j})_{\boldsymbol{z}} \\ (\boldsymbol{W_j})_{\boldsymbol{m}} \end{bmatrix} \cdot \frac{\vec{ij}}{\|\vec{ij}\|},$$

(Eq. 3.3)

where i is the ignited polygon's site, j is the neighbouring polygon's site, $(W_{ij})_n$ is the normal wind from i to j (m/s), $(W_j)_z$ is the mean zonal (West-to-East) wind speed of j (m/s), $(W_j)_m$ is the mean meridian (South-to-North) wind speed of j (m/s), and $\vec{ij}$ is the vector from polygon site i to polygon site j.

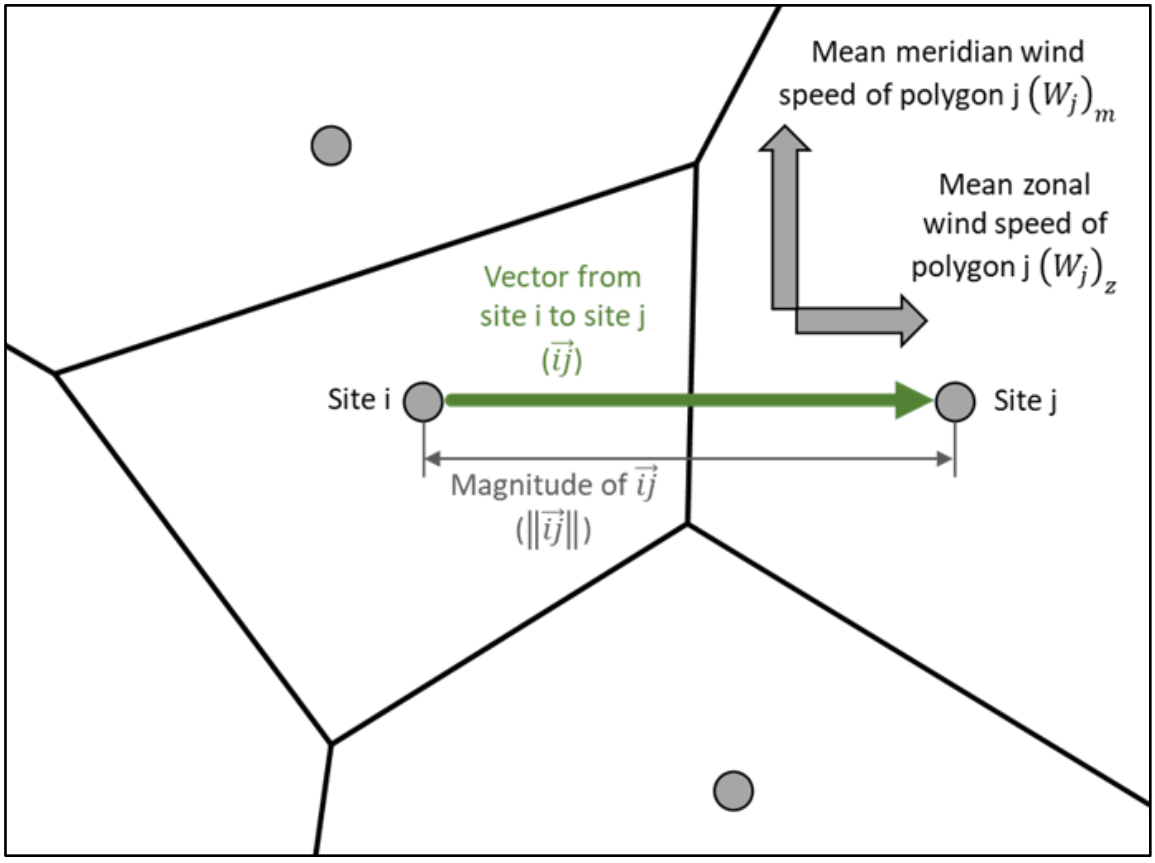

Figure 3.6: Finding the normal wind for fire spreading from site i to j by getting the dot product of both the mean zonal and meridian wind speed properties of polygon j and the unit vector of propagation from site i to j.

A visualisation was created to show the spatial distribution of the polygons and evolution of the fire line. The polygons were rendered in distinct colours to represent their states with red representing polygons that were currently on fire, black representing those that were burnt, clear without colour representing those that had not caught fire yet and green

representing wildfire source polygon(s), where the fire started. The colour of a polygon was determined from a combination of the masses in the states mentioned. A 50% transparency was used to allow for the underlying map data to be seen (Figure 3.7).

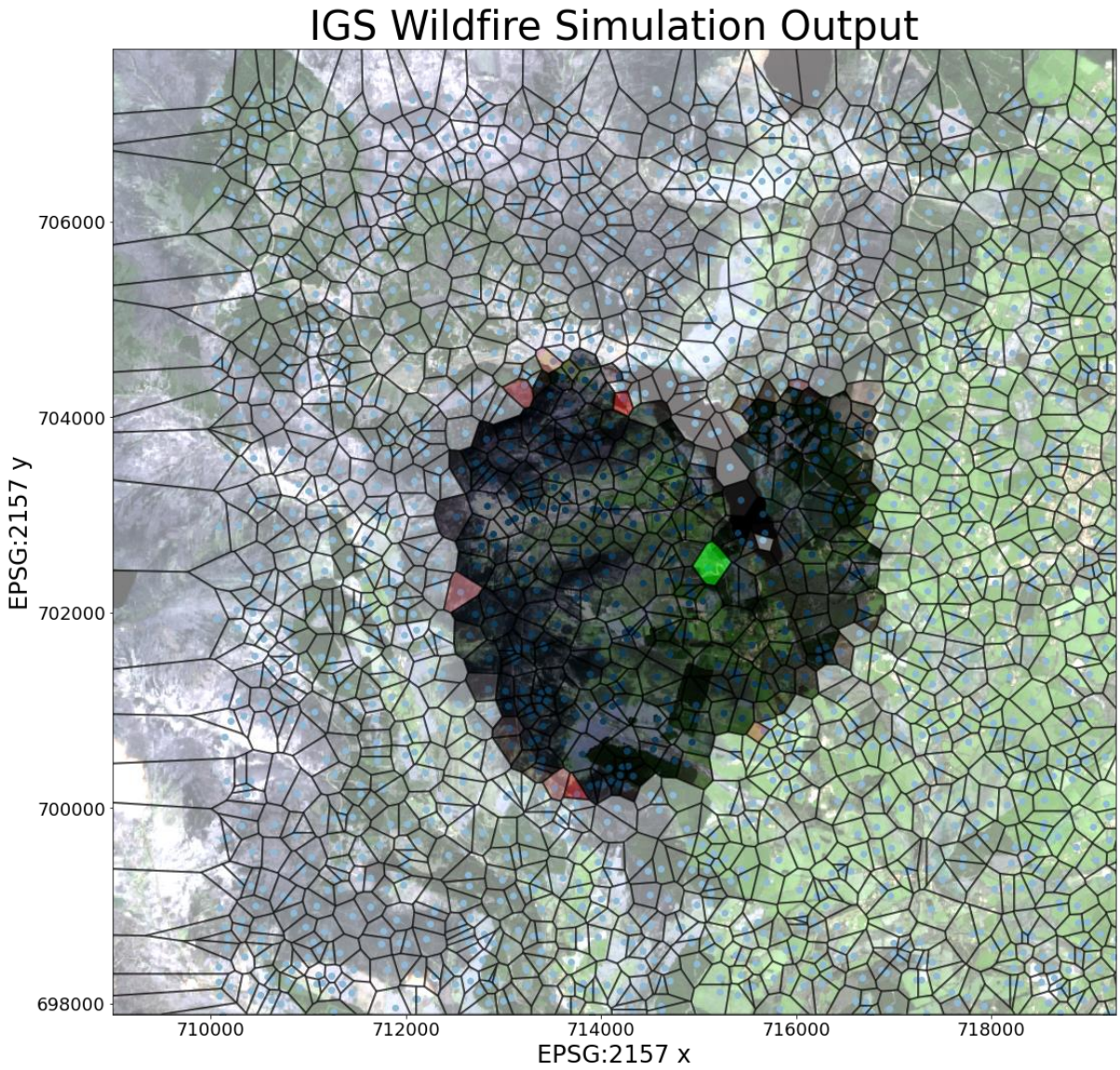

Figure 3.7: Sample simulation output after 100,000 seconds showing not burnt (white and transparent), wildfire source (green), burning (red), and burnt (black) polygons based on the mass of the polygon in that state, with a coloured satellite image of the terrain in the background (Copernicus 2022).

The area of the polygon was found in pixel coordinates and converted to an area in m^2. The oven-dry fuel load per m^2 was sourced from the "fuels file" and used to estimate the total fuel per polygon (kg). The total fuel mass in the polygon could then be split into flammable vegetation, currently ignited vegetation and vegetation that had been completely burnt, written as:

$$V_i = N_i - (I_i + B_i),$$

(Eq. 3.4)

where i is a polygon, $N_i \geq 0$ is the total amount of fuel in polygon i (kg), $V_i \geq 0$ is the amount of remaining flammable vegetation in polygon i (kg), $I_i \geq 0$ is the amount of fuel in polygon i that is currently ignited (kg), and $B_i \geq 0$ is the amount of fuel in polygon i that has burnt (kg).

To estimate the fuel mass of a polygon that was currently on fire, the propagation ratios from the ignited polygon and its neighbours were found. The propagation ratios of fire that had spread within the ignited polygon (both from that polygon and to that polygon) and the respective distances, between the ignited and neighbouring polygons were multiplied to find the total distances fire had spread within the ignited polygon. This was then divided by the sum of half of each distance between the ignited and neighbouring polygons to get an estimate for the proportion of the ignited polygon that has caught fire. In simpler terms, the numerator was the distance fire had spread within the ignited polygon, and the denominator was the half distances between the ignited polygons and its neighbours. This created a dimensionless metric which scaled from 0 to 1. The proportion of the ignited polygon that had caught fire was multiplied by the total amount of fuel in the polygon to find the total amount fuel that had ignited. The amount of completely burnt fuel was then subtracted producing the fuel mass that was currently on fire:

$$q_{ij} = \left(P_{ij}\right) + \left(max\{P_{ji}, 0.5\} - 0.5\right),$$

(Eq. 3.5)

$$I_i = N_i \left(\frac{\sum_{j=1}^{n_i} min\{q_{ij}, 0.5\} d_{ij}}{\sum_{j=1}^{n_i} \frac{d_{ij}}{2}} \right) - B_i,$$

(Eq. 3.6)

where i is a polygon, j is a polygon neighbouring i, $q_{ij} \geq 0$ is the total propagation ratio of fire that has spread within polygon i between it and polygon j, $N_i \geq 0$ is the total amount of fuel in polygon i (kg), $I_i \geq 0$ is the amount of fuel in polygon i that is currently ignited (kg), $B_i \geq 0$ is the amount of fuel in polygon i that has burnt (kg), $0 \leq P_{ij} \leq 1$ is the propagation ratio from polygon i to j, $0 \leq P_{ji} \leq 1$ is the propagation ratio from polygon j to i, $n_i \geq 0$ is the number of neighbours polygon i has and $d_{ij} > 0$ is the distance between polygons i and

j (m). Please note if $q_{ij} > 0.5$ then the value of q_{ij} is set to 0.5 and if P_{ji} < 0.5 then the value of P_{ji} is set to 0.5. This was to ensure only fire that had spread within polygon i was measured as any other spread in neighbouring polygons was used to find the mass of their polygon that was on fire instead.

The reaction intensity parameter from the Rothermel model found the energy produced per unit area per time step. Using the equation for the reaction intensity, the energy produced per unit mass per unit time step (E_{ij}) can be evaluated. This is done by removing the net fuel load $(w_n)_j$ from (Eq. 3.7) (which is (Eq. 2.12) formatted to work with IGS polygons) to give (Eq. 3.8) which, as previously stated, uses United States customary units. $E_{ij}$was found using the fuel properties from neighbouring polygons and was converted to metric units.

$$(I_R)_j = \Gamma'_j \, (w_n)_j h_j \eta_{M_j} \eta_{S_j},$$

(Eq. 3.7)

$$E_{ij} = \Gamma'_j \, h_j \eta_{M_j} \eta_{S_j},$$

(Eq. 3.8)

where i is a polygon, j is a polygon neighbouring i, $(I_R)_j \geq 0$ is the reaction intensity of polygon j in J/m^2/s (Btu/ft^2/min), $(w_n)_j \geq 0$ is the net fuel load of polygon j in kg/m^2 (lb/ft^2), $E_{ij} \geq 0$ is the energy produced spreading from polygon i to polygon j per mass per time increment in J/kg/s (Btu/lb/min), $\Gamma'_j \geq 0$ is the optimum reaction velocity of polygon j in s^{-1} (min^{-1}), $h_j \geq 0$ is the low heat content of polygon j in J/kg (Btu/lb), $0 \leq \eta_{M_j} \leq 1$ is the moisture damping coefficient of polygon j and $0 \leq \eta_{S_j} \leq 1$ is the mineral damping coefficient of polygon j. Just like the base Rothermel model, (Eq. 3.8) may have poorer performance in some fuel beds at lower fuel loadings, especially those with high packing ratios. Minor modifications have been made to the Rothermel model that helped reduce the oversensitivity to fuel depth in the past [149]. However, this updated form of the Rothermel model was not used by the IGS in this chapter as ForeFire used the base Rothermel model which would make any comparison unfair.

The mean energy produced from burning fuel while the fire spread to its neighbouring polygons was calculated by multiplying the mean energy produced (E_{ij}) from all neighbouring polygons with the mass of ignited fuel in the polygon and the time step. The

energy was then divided by the low heat content of the polygon to get an estimation of the amount of ignited mass lost to burning:

$$dB_i = \frac{\frac{\sum_{j=1}^{n_i} E_{ij}}{n_i} I_i dT}{h_i},$$

(Eq. 3.9)

where i is a polygon, j is a polygon neighbouring i, $E_{ij} \geq 0$ is the energy produced spreading from polygon i to polygon j per mass per time increment in J/kg/s (Btu/lb/min), $dB_i \geq 0$ is the rate at which ignited fuel in polygon i becomes burnt fuel in kg/dT, $n_i > 0$ is the number of polygons that neighbour i, $I_i \geq 0$ is the amount of fuel that is currently ignited in polygon i in kg, $dT > 0$ is the time step in s and $h_i \geq 0$ is the low heat content of polygon i in J/kg (Btu/lb). This value was then added to the total amount of fuel that has been burnt in that polygon:

$$(B_i)_{t+dT} = (B_i)_t + dB_i,$$

(Eq. 3.10)

where i is a polygon, $dT > 0$ is the time step (s), $dB_i \geq 0$ is the rate at which ignited fuel in polygon i becomes burnt fuel (kg/dT), t is time in the simulation (s), and $(B_i)_t \geq 0$ is the amount of fuel that is burnt in polygon i at t time (kg). Tracking both ignited and burnt fuel amounts paints a clear image of what areas are currently on fire and which have ceased burning due to a lack of available flammable vegetation (Figure 3.7).

The collection of polygons which had fire reach their sites can be converted into a fire line, representing the outer perimeter of the fire. This allows for the IGS to be compared to other continuous programs such as ForeFire. The boundary polygons need to be found to produce a fire line. A straightforward algorithm was developed to find the boundary polygons where each polygon was checked to see if it can spread the fire (a polygon is only able to spread fire if fire from the model has reached that polygon's site). Once a set of polygons capable of spreading fire is found, it can be shortened by checking if these polygons have one or more neighbours that cannot spread fire. This gives an updated set of boundary polygons that are on the fire line perimeter of the IGS. This was then converted to a list of boundary edges by checking the edges of each boundary polygon and recording any edges between a boundary polygon and a polygon that cannot spread fire. Getting the boundary edges of the boundary

polygons helps reduce underpredicting how far the fire has spread within a given boundary polygon. Polygons that can spread fire while also touching the outer perimeter of the NetCDF file are checked, and any edges of that polygon on the outer perimeter were added to the list of boundary edges. Edges each consist of two vertices. To begin a random edge is selected from the list of boundary edges and a recursive algorithm finds the next edge to share a vertex with the randomly selected edge, removing it from the list and storing it in a stack. This process is repeated until no more shared vertices can be found, and the stack then becomes an ordered boundary. The process of selecting a random edge continues until all edges have been removed from the list (Figure 3.8). Sometimes the vertices of edges do not line up perfectly, so a 3m (approximately 0.01 % of the region width) margin of error between vertices of different edges was allowed to ensure a completed shape. All edges shorter than the margin of error were removed to prevent inaccuracies generating the completed boundary.

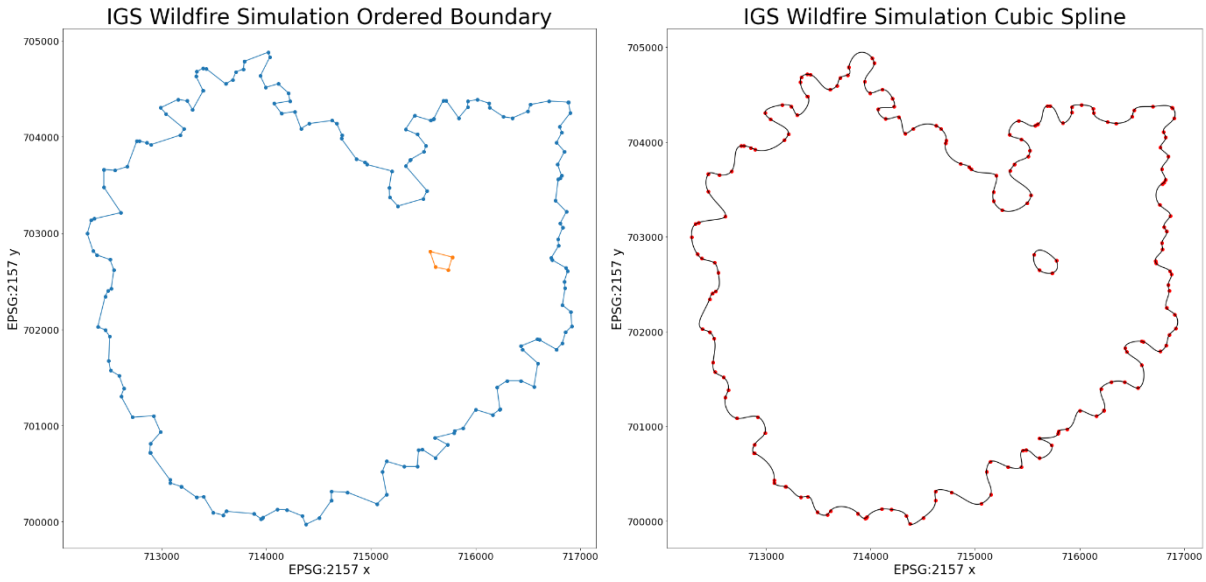

Figure 3.8: Left: Ordered boundary consisting of polygon edges from the IGS output that create an outer perimeter of the simulated wildfire. Right: the ordered boundary smoothed by a cubic spline.

The ordered boundary was then smoothed to provide the final representation of the fire line, using a cubic spline. The SciPy Python library has a subpackage called “interpolation” [150]. This subpackage can be used to generate cubic splines given separate lists of x and y coordinates. Traditionally cubic splines are performed where x coordinates are ordered in increasing value. This will not work for the IGS’s output as reordering the x coordinates would

make the boundary unordered and ruin the boundary shape. To circumvent this, a parameter (c) was defined as the cumulative distance between all previous points in the ordered boundary. For example, the first point would have c equal to 0, while the second point would be equal to the distance between the first and second point. The third point would then have c equal to the distance between the first and second point summed with the distance between the second and third point. This pattern continues for all points in the ordered boundary. Due to c being a list of increasing values, it was possible to perform a cubic spline interpolation on both the x and y coordinates separately using c [151]. The x and y coordinates can then be re-combined to produce a cubic spline. With the cubic spline, the fire line becomes a smooth curve, more closely resembling how a real fire would appear (Figure 3.8).

3.3 Methodology to Compare Two Wildfire Outputs

A new method was developed to compare the similarity of outputted wildfires. This method took inspiration from both set theory and receiver operating characteristics (ROCs). When comparing outputted wildfires, one of the wildfires was treated as the ground truth. For this chapter ForeFire was selected as the ground truth due to its increased resolution; therefore, the IGS was selected as the predicted wildfire. The areas of predicted fire that overlapped with the ground truth were treated as true positives. Areas where the IGS predicted fire and ForeFire did not were false positives. Areas where ForeFire predicted fire and the IGS did not were false negatives. Areas where both the IGS and ForeFire predicted there would not be fire were true negatives (Figure 3.9). The area of true negatives could be considered infinite, dependent on the size of the plane where the fire is being simulated. Therefore, true negatives were not counted and any confusion matrix calculations involving them were not used.

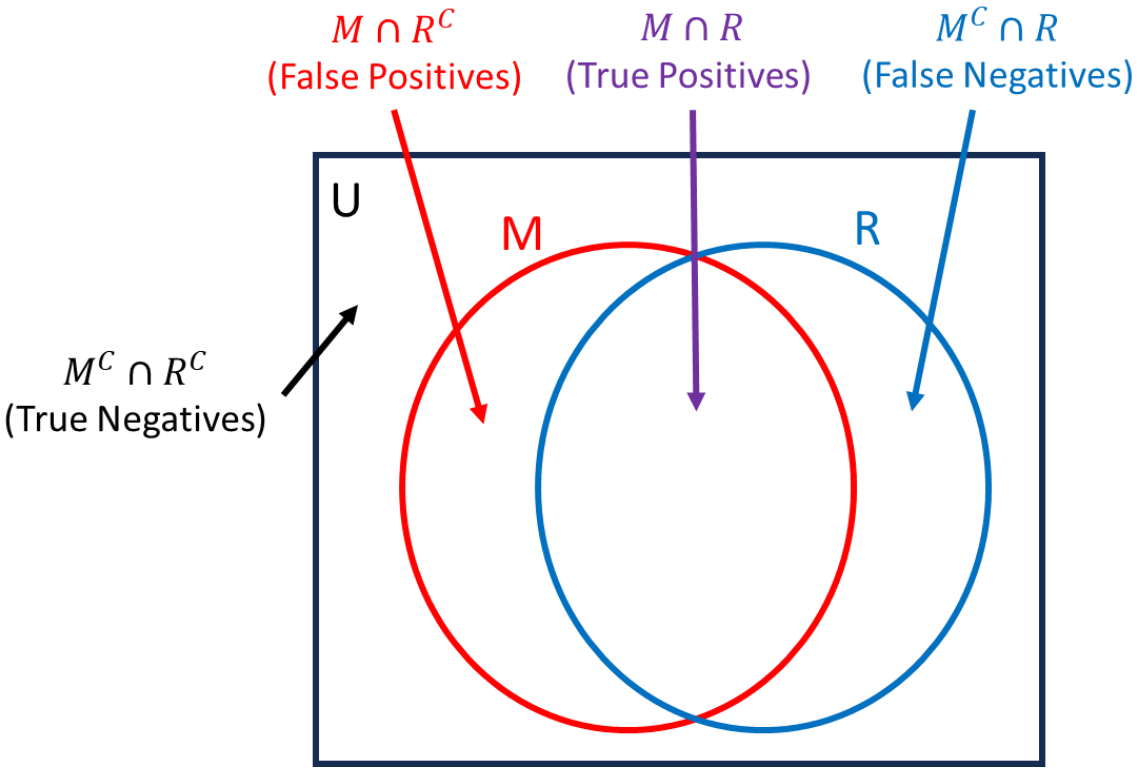

Figure 3.9: Calculating similarities of the IGS (M) and ForeFire (R) wildfire shape outputs, where the areas of $M \cap R$ are true positive, that of $M \cap R^C$ are false positive, that of $M^C \cap R$ are false negative, and that of $M^C \cap R^C$ are true negative.

The threat score was chosen as the metric of accuracy due to it measuring true positives, false positives, and false negatives simultaneously without requiring a measurement of true negatives. The default threat score equation was used but two weighting factors were added to allow for the scaling of how important false positives and false negatives are relative to the true positives. It is given by:

$$TS = \frac{TP}{TP + w_1 FN + w_2 FP},$$

(Eq. 3.11)

where TS is the threat score, TP is the area of true positives, FN is the area of false negatives, FP is the area of false positives, $w_1 \geq 0$ is the weighting coefficient for FN and $w_2 \geq 0$ is the weighting coefficient for FP.

If TS is 1, both the IGS and ForeFire polygons are a perfect match, while if TS is 0, there is no overlap between the IGS and ForeFire fire lines. Increasing w_1 or w_2 will make FN and FP respectively have a greater impact on lowering TS, while decreasing w_1 or w_2 will have the opposite effect. For the comparison of results in this chapter $w_1 = w_2 = 1$.

3.4 Comparison of Grid Types

Five different grid types were compared to find which grid type may be the optimal grid for comparison with ForeFire. All five grids were built using Voronoi diagrams. The five grid types consisted of randomly positioned sites; triangular tessellating grid; square tessellating grid; hexagonal tessellating grid; and a grid where more sites would be positioned closer to the wildfire source and on land cover types defined by the fuel properties lookup table, which are important to monitor. This grid is referred to as the focused grid. Each grid uses the von Neuman neighbourhood for determining which cells neighbour other cells (corner neighbouring cells exclusive) [57].

The grid with randomly positioned sites takes as input the number of sites it must position and a bounding box to position the sites inside. It randomly positions that number of sites within the bounding box. If a site is positioned at the same coordinates as an existing site, then that site will be re-positioned. Each polygon in the randomly positioned grid will have at least one neighbour ranging to the total number of polygons minus one, potential neighbours (Figure 3.10).

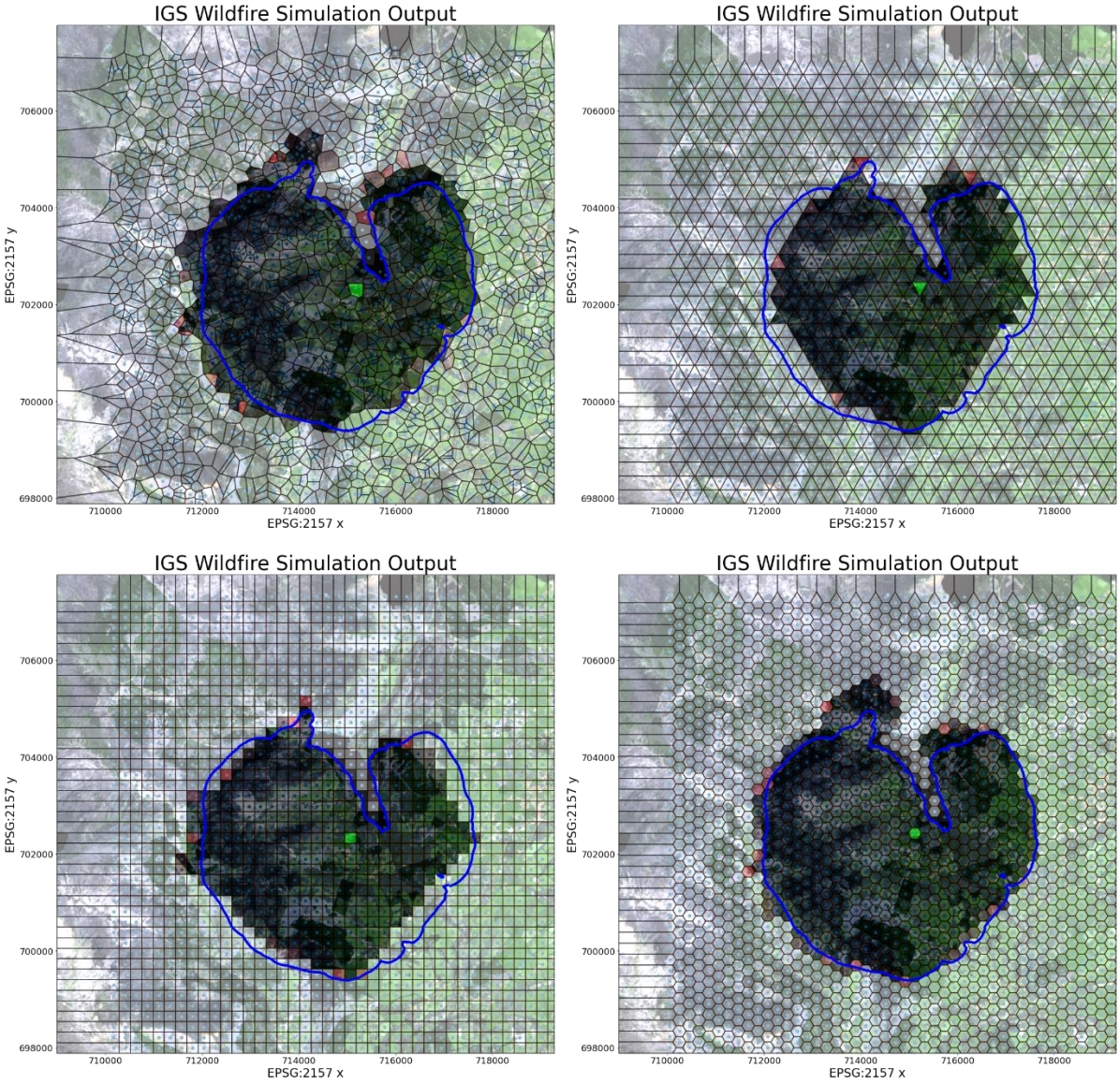

Figure 3.10: Outputs of different IGS grid types (coloured polygons) compared to ForeFire's output (blue polygon). On the top left are randomly positioned sites, on the top right is a triangular grid, on the bottom left is a square grid, and on the bottom right is a hexagonal grid (Copernicus 2022).

The triangular grid consists of tessellating equilateral triangles of equal size. It takes as input the number of sites to position and a bounding box to position the sites inside. It is one of the more complicated grid types to position using a Voronoi diagram. The program iterates through possible different numbers of sites to place on each row. It does this until it finds the smallest number of sites to place in a row, where once the entire grid is filled vertically there will still be approximately the same number of sites as originally inputted. Using these values, it is possible to find the horizontal distance between sites and therefore the required vertical height of triangles on each column using the properties of equilateral and right-angled triangles, through the following equation:

$$y = \sqrt{3x^2} = Y_l + Y_s,$$

(Eq. 3.12)

where $y > 0$ is the height of the equilateral triangles, $x > 0$ is the horizontal distance between sites, $Y_s > 0$ is the short vertical offset, and $Y_l > 0$ is the long vertical offset. The long and short vertical offset were used to alternate the orientation of triangles which is required for tessellation. The short vertical offset is calculated using the principle that sites for equilateral triangles should be equidistant from all corners of the triangle. The long vertical offset can be found using the short vertical offset. Equations showing this are as follows:

$$Y_s = -\frac{x^2 - y^2}{2y},$$

(Eq. 3.13)

$$Y_l = \sqrt{x^2 + Y_s^2},$$

(Eq. 3.14)

where $x > 0$ is the horizontal distance between sites, $y > 0$ is the height of the equilateral triangles, $Y_s > 0$ is the short vertical offset, and $Y_l > 0$ is the long vertical offset. In triangular tessellation on a singular row the triangles appear upright (generated from the short vertical offset) followed by an upside-down triangle in a repeating pattern. The upside-down triangles require the long vertical offset to be placed correctly. Rows and columns that are both indexed even or odd have the short vertical offset is applied. While on odd-indexed rows and even indexed columns, or even-indexed rows and odd indexed columns, the long vertical offset is applied (Figure 3.11). Due to rounding errors in Python and the Foronoi Python library, additional small edges were sometimes created between sites. To fix this, if the length of an edge was under a certain threshold it was removed to prevent the grid from assuming some sites had additional neighbours. Each polygon in the triangular grid (excluding the edge) will have three neighbours (Figure 3.10).

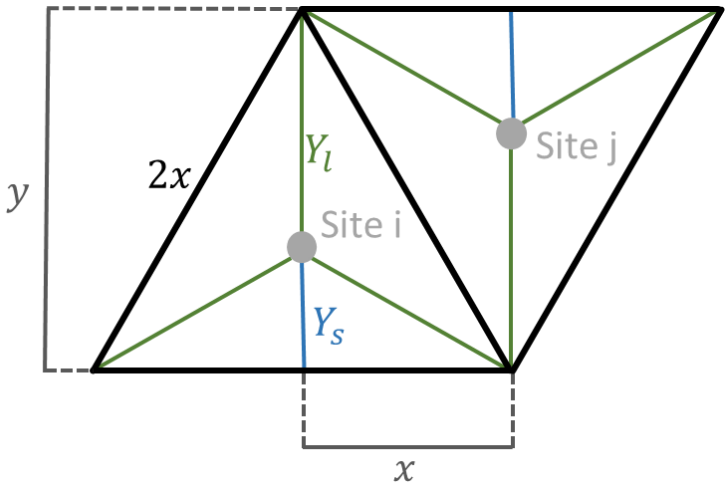

Figure 3.11: Calculation of Voronoi diagram site placement for equilateral triangular tessellation. Given the distances of x and y, Y_s and Y_l can be derived using the Pythagorean theorem.

The square grid consists of tessellating squares in a chessboard pattern, where the squares are of an equal size and aligned both horizontally and vertically. The input consists of the number of sites and a bounding box to position the sites inside. The floored square root of the inputted number of sites is found and used to equally space sites within the bounding box into equidistant columns and rows. Each polygon in the square grid (excluding the edge) will have four neighbours (Figure 3.10).

The hexagonal grid consists of tessellating regular hexagons (flat sides on top and bottom) of an equal size defined by the inputted number of sites and a bounding box. Like the triangular grid, the program iterates through the possible different numbers of sites to place on each column until it finds the optimal number. The optimal number of sites to place is found by getting a number as close to what was originally stated within the bounding box with a calculated horizontal and vertical distance between sites. Every even column is then offset vertically by half the vertical distance between sites. The horizontal distance between the sides was found as a function of the vertical distance using the Pythagorean theorem (Figure 3.12). A simplified version of the expression was then found:

$$x = \frac{\sqrt{3y^2}}{2},$$

(Eq. 3.15)

where $x > 0$ is the horizontal distance between sites and $y > 0$ is the vertical distance between sites. Each polygon in the hexagonal grid (excluding the edge) will have six neighbours (Figure 3.10).

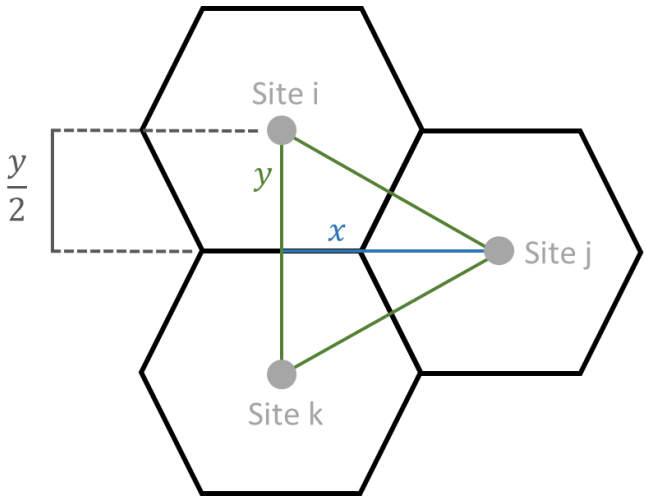

Figure 3.12: Calculation of Voronoi diagram site placement for regular hexagon tessellation. Given the distances of y, x can be found using the Pythagorean theorem.

A random grid with increased focus on a region of interest determined by risk assessment was created and referred to as the focused grid. It was designed to have a greater density of sites in regions most likely to be affected by the wildfire and regions that are important to monitor. The regions that are most likely to be impacted the most by the wildfire tend to be situated closer to the ignition point of the fire or contain land cover types that can sustain

large fires (e.g. forested areas). Areas of preselected importance tend to be areas with a higher density of people (e.g. urban areas). The program runs through each pixel of the land cover map within the bounding box where it has a base probability of selecting the pixel it is currently on (this ensures that not too many sites are generated). Once a pixel has been selected a second probability is generated to determine if a site should be placed at this position based on the distance from the ignition point and the land cover type of that pixel:

$$G = F(1 + m)^{-k},$$

(Eq. 3.16)

where $0 \leq G \leq 1$ is the probability of a site being placed, $0 \leq F \leq 1$ is the coefficient affecting the probability of site placement based on land cover type, $m \geq 0$ is the distance between the current pixel and the mean ignition point (found by getting the mean coordinates of all polygon sites, for polygons that contain ignition points) divided by the distance between the mean ignition point and furthest point within the bounding box, and $k > 0$ is a coefficient that weighs site placement probability based on $(1 + m)$. This allows for most of the computing to be performed where it is likely to be needed instead of wasting resources in areas the fire is unlikely to reach or to spread fast in. Each polygon in the focused grid will have at least one neighbour ranging to the total number of polygons minus one, potential neighbours (Figure 3.13).

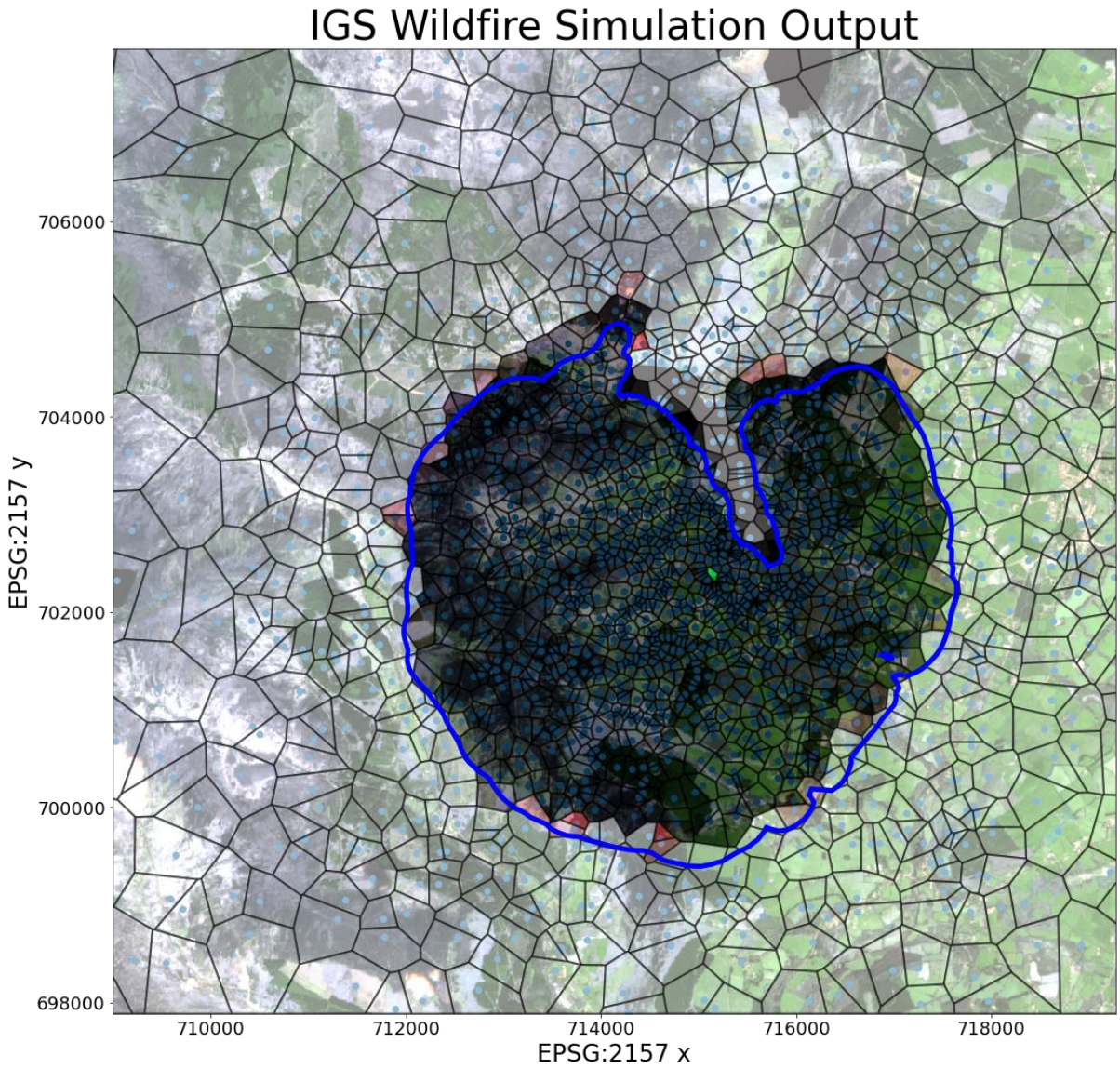

Figure 3.13: Output of the focused grid (coloured polygons) compared to ForeFire's output (blue polygon) (Copernicus 2022).

A fire was set at the coordinates EPSG:2157 (715122, 702388) at 0 seconds in the simulation. The environment had no wind, and the fire was simulated for 100,000 seconds with a time step of 1,000 seconds. The fire was simulated 10 times for each grid type, where the time it took for the IGS to simulate the fire and the threat score when compared to a ForeFire simulation were both recorded (Table 3.1). Values for the equation of site placement of the focused grid were selected so the focused grid would have approximately 2,000 sites each time.

Grid Types	Triangular	Square	Hexagonal	Random		Focused		
Number of Sites	1952	1936	1974	2000		Variable (See Below)		
Threat Score	0.84	0.84	0.82	Variable (See Below)		Variable (See Below)		
	Simulation Time (s)	Simulation Time (s)	Simulation Time (s)	Simulation Time (s)	Threat Score	Number of Sites	Simulation Time (s)	Threat Score
Fire 1	32.8	31.65	45	46.71	0.84	1995	52.46	0.87
Fire 2	33.27	31.56	43.43	49.99	0.79	1975	52.32	0.86
Fire 3	32.05	31.82	43.76	44.35	0.78	1980	52.37	0.85
Fire 4	31.86	32.14	44.03	48.83	0.82	1946	50.26	0.91
Fire 5	32.26	32.08	44.41	43.85	0.8	2032	54.88	0.86
Fire 6	32.19	32.14	43.4	44.7	0.72	2034	53.04	0.89
Fire 7	32.02	31.81	44.3	45.52	0.79	2044	53.14	0.89
Fire 8	33.86	32.63	44.1	42.06	0.83	1996	52.46	0.83
Fire 9	31.89	31.41	44.13	40.8	0.76	2002	52.28	0.89
Fire 10	32.02	31.93	44.29	40.9	0.79	1973	50.96	0.9
Mean	32.42	31.92	44.08	44.77	0.79	1998	52.42	0.87
Standard Deviation	0.67	0.35	0.47	3.11	0.03	31	1.24	0.03
Standard Error	0.21	0.11	0.15	0.98	0.01	10	0.39	0.01

Table 3.1: Number of sites, threat scores and simulation times for the different grid types. All grid types excluding the focused grid have a constant number of sites. All grid types except the random grid and focused grid have a constant threat score.

The regular grids and the random grid had a set number of sites while the focused grid had a varying level of sites due to the nature of its site generation. As the regular grids showed no variability in their grid layout, each simulation produced the same wildfire. Therefore, their threat score was constant throughout, unlike the random grid and focused grid. The regular grids tend to have quicker execution times than the random grid and focused grid, where the focused grid has the longest simulation time. The random grid and focused grid also have significantly higher standard deviations and standard errors in their simulation time; this is due to both grids generating a completely new grid layouts each time the simulation is run, while the regular grids always have the exact same grid layout. The random grid had the lowest threat score, while the focused grid had the highest threat score.

The triangular, square, and hexagonal grids can estimate the shape of the fire line well as shown by their threat scores, but they are confined to their regular shape and may struggle to represent some of the more irregular shapes of actual fire lines. The regular grids all share the property of polygons having an equal number of neighbours (excluding at the bounding box edges) that are all equidistant from each other, which gives all polygons an equal spreading opportunity (Figure 3.10). The regular grids had an equal resolution across the entire grid. This leaves room for further optimisation where areas less likely to burn according to the model could have a lower resolution.

The randomised Voronoi grid can produce some of the more irregular shapes, which can allow it to follow the irregular pattern of real terrain. Unlike the regular grids each polygon does not have an equal spreading opportunity due to the distance between neighbouring sites varying. The randomised Voronoi grid has a similar resolution throughout the grid which was not an optimal use of computation power. The focused grid can follow the shape of the fire line with greater precision due to its high resolution in areas of importance. It also provides flexibility of where in the scene computation occurs. It does not have an equal spreading opportunity for each polygon and takes longer than all other types of grids to simulate the spread of fire (Figure 3.13).

The focused grid from the IGS was chosen for comparison with the industry standard wildfire modelling program ForeFire due to its high threat score and reasonable simulation time.

3.5 Summary of Results

To compare the IGS and ForeFire, 20 wildfires were simulated. The wildfires were randomly placed in various locations within the Wicklow Mountains, Ireland with varying wind speeds and for different durations. This was done to test the versatility of the IGS. Some simulations included multiple separate wildfires. Outputs of ForeFire and the cubic spline of the IGS using the focused grid were graphed and then their similarity was found (Figure 3.14). For each of the simulations performed using the IGS, the time step used was 1% of the simulation duration, with a minimum time step of 1 second. While simulating these

wildfires the time it took to run the simulation of the IGS usually took longer than ForeFire, even though it required less computing. This was due to ForeFire being built in the more computationally efficient language C++ [152]. To make the comparison between the IGS and ForeFire fair, the simulation loop of the IGS was ported to C++ from Python. The remaining code for the IGS was still in Python due to easier accessibility of dependent libraries. The comparison does not include the setup times for ForeFire or the IGS.

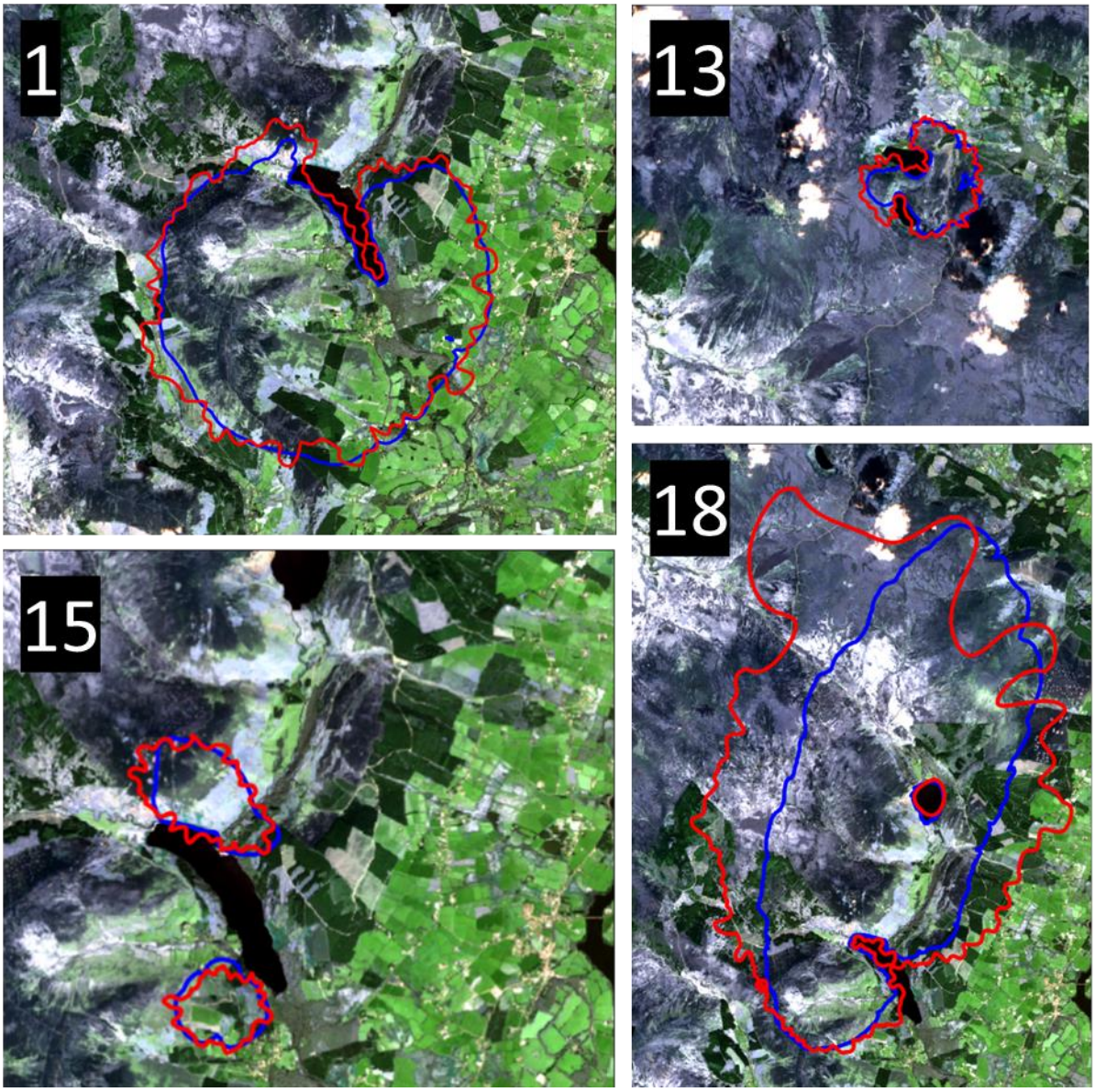

Figure 3.14: IGS outputs (red polygon) compared to ForeFire output (blue polygon). (Top left: Wildfire 1, Top Right: Wildfire 13, Bottom left: Wildfire 15 and Bottom Right: Wildfire 18 (Table 3.2)). The IGS outputs tend to extend beyond the ForeFire outputs, this is most likely due to the IGS selecting the outer boundary edges of boundary polygons when creating the ordered boundary to reduce underfitting. (Copernicus 2022).

The time it took to run the simulation loops of ForeFire and the IGS in both Python and C++ were recorded and compared. The areas of true positives, false positives, and false negatives for the IGS when compared to ForeFire were recorded for each wildfire, and the corresponding threat score were calculated. These data are presented in Table 3.2. Histograms of the threat scores and IGS runtime in C++ were generated to visualise the results (Figure 3.15).

From the sample wildfires simulated, the IGS runs on average 34 times more quickly than ForeFire with a mean fire line similarity of 0.8 when compared to ForeFire. The use of the IGS instead of ForeFire allows for multiple wildfires to be simulated (restricted to one area) in the same amount of time it would take ForeFire to run a simulation of one wildfire. From Table 3.2 it is evident that the fire lines produced by the IGS when compared to their ForeFire counterpart tended to contain mostly areas of true positives followed by false positives and finally false negatives. There is also a trend where wildfires simulated by the IGS that run for longer durations tend to not perform as well as those run for shorter durations relative to ForeFire. There are some limitations with the IGS. The focused grid may lose precision in larger wildfires as the frequency of polygons decreases the further a wildfire spreads from its ignition point. This is evident in Wildfire 18 (Figure 3.14), where there is a large drop in site resolution at distances far away from the wildfire ignition point. The large distance between the IGS and ForeFire's northern and western fire lines highlights this issue. The cubic spline forces the curve to intersect the points on the edges of polygons. Due to the resolution of sites within the IGS, this makes the cubic spline produce small modulations on the fire line, while ForeFire has straighter edges.

Wildfire	Start Points and Time (x, y, t) (EPSG:2157, s)	Wind (Zonal, Meridian) (m/s)	Simulation Duration (s)	ForeFire Time (s)	IGS Python Time (s)	IGS C++ Time (s)	ForeFire - IGS C++ Time (s)	ForeFire / IGS C++ Time	True Positives (m^2)	False Positives (m^2)	False Negatives (m^2)	Threat Score
1	(715122, 702388, 0)	(0, 0)	100,000	5.3	46	0.193	5.105	27	21,162,353	2,597,119	594,021	0.87
2	(713350, 704946, 0)	(0, 0)	100,000	7.1	58	0.321	6.773	22	31,641,462	6,709,658	393,613	0.82
3	(715651, 706392, 0)	(0, 0)	100,000	6.9	68	0.267	6.613	26	29,336,486	6,015,675	477,550	0.82
4	(710773, 707941, 0)	(0, 0)	100,000	5.9	72	0.294	5.584	20	28,808,953	4,668,653	492,470	0.85
5	(713513, 708268, 0)	(0, 0)	100,000	6.6	61	0.256	6.342	26	28,944,866	6,508,166	543,992	0.80
6	(712496, 708990, 0)	(0, 0)	100,000	5.9	65	0.314	5.561	19	28,969,212	4,046,505	273,223	0.87
7	(715823, 708190, 0)	(0, 0)	100,000	7.1	61	0.289	6.791	24	31,247,060	5,138,147	315,410	0.85
8	(713467, 711230, 0)	(0, 0)	100,000	6.3	63	0.291	6.018	22	29,105,672	3,749,638	351,839	0.88
9	(714378, 701907, 0)	(0, 0)	100,000	5.5	55	0.238	5.263	23	25,224,112	2,781,007	1,005,543	0.87
10	(704110, 703419, 0)	(0, 0)	100,000	7.2	55	0.257	6.896	28	30,443,340	3,596,271	892,268	0.87
11	(709231, 705966, 0)	(0, 0)	50,000	1.8	27	0.072	1.717	25	5,684,170	418,904	771,298	0.83
12	(714122, 706378, 0)	(0, 0)	50,000	2.5	26	0.081	2.451	31	6,941,426	482,771	772,200	0.85
13	(714027, 716143, 0)	(0, 0)	30,000	1.4	15	0.028	1.394	51	1,701,109	497,978	84,778	0.74
14	(713369, 707072, 0)	(0, 0)	120,000	9.9	75	0.371	9.509	27	43,051,154	8,672,817	792,791	0.82
15	(715122, 702388, 0)	(0, 0)	20,000	1.7	11	0.043	1.621	39	1,833,936	293,936	237,246	0.78
	(714868, 704531, 2000)											
16	(715148, 702437, 0)	(0, 0)	30,000	2.2	14	0.034	2.142	64	3,856,323	398,370	671,296	0.78
	(715930, 703049, 1200)											
	(715804, 702092, 2700)											
17	(714633, 704695, 0)	(0, 0)	50,000	3.7	27	0.106	3.609	35	13,747,087	3,459,420	326,943	0.78
	(715297, 707714, 0)											
18	(713945, 703907, 0)	(1, 3)	50,000	10.6	31	0.158	10.404	67	42,742,800	25,938,603	3,724,274	0.59
19	(712649, 715686, 0)	(1, -3)	20,000	2.1	11	0.040	2.068	53	4,779,324	1,406,943	1,845,529	0.60
20	(713925, 707395, 0)	(-0.5, -0.4)	30,000	2.0	17	0.043	1.997	47	4,085,839	283,454	714,927	0.80
Mean				5.1	43	0.185	4.893	34	20,665,334	4,383,202	764,061	0.80
Standard Deviation				2.8	23	0.118	2.664	15	13,902,996	5,658,324	792,456	0.08
Standard Error				0.6	5.1	0.03	0.6	3.3	3,108,804	1,265,240	177,199	0.02
Sum									413,306,686	87,664,034	15,281,210	

Table 3.2: Comparison of 20 simulated wildfires in ForeFire and the IGS under the metrics of computing time (ForeFire/IGS C++ Time) and similarity of the resulting fire line (Threat Score, where $w_1, w_2 = 1$).

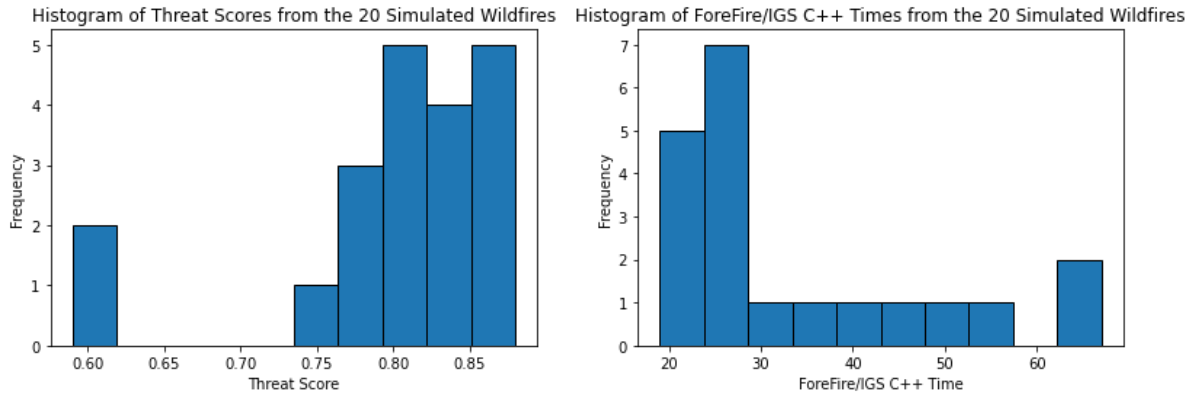

Figure 3.15: Histograms with values split into 10 bins of the threat scores and ForeFire/IGS C++ times from the 20 simulated wildfires in Table 3.2.

3.6 Discussion

Comparing the five different grid types produced a table of computational times and the similarity of their outputs, relative to ForeFire (Table 3.1). The regular grids had no variability in their grid layout. They also took approximately the same time to run the simulation as shown by the small standard deviation in their computational times. The small standard deviation can be accounted for with other uncontrollable factors affecting the computer such as background tasks. Both irregular grids have a much higher standard deviations in their computational times. This is because the process of seeding these grids is random, so a different grid is produced each time a simulation is run. Some of the irregular grids have overall regions of higher resolution than others, which can lead to varying computational times. The number of neighbours each grid has is also correlated to the computational time, as the program must calculate the propagation ratio between each ignited polygon that can spread fire to its neighbours. The square and triangular grids have the fastest computational times while having four and three neighbours respectively. The hexagonal grid has the next fastest computational time with six neighbours. The random grid has a computational time like that of the hexagonal grid even though there is no set number of neighbours per polygon. This is most likely due to each polygon having approximately six neighbours. The focused grid has the slowest computational time, even though most polygons also have approximately six neighbours. This is due to the increased density of polygons close to the wildfire source, as propagation ratios need to be calculated for all these ignited polygons and their neighbours that fire can spread to. Before the

simulation loop, polygons in all grid types have a list of their neighbouring polygons saved. This can take up a large amount of memory but lowers computation time during the simulation loop. If neighbouring polygons were calculated in real time during the simulation loop, irregular grids may have even longer computation times than regular grids, as polygons in irregular grids do not follow a straightforward indexing method. In this simulation all grids use the same indexing approach so a comparison could be made.

In terms of the output similarity to ForeFire, the focused grid performs the best followed by the square, triangular, hexagonal, and random grids. As previously mentioned, the focused grid has a high density of polygons near the wildfire source, allowing for the simulation to nearly follow the continuous movement of the wildfire simulated in ForeFire. For the regular grids to be comprised of regular shapes it was only possible to have a set number of polygons in the simulation area. In Table 3.1, the number of sites for the regular grids was set as close to 2,000 as possible while still retaining a regular grid. The irregular random grid tends to produce outputs with the lowest similarity to ForeFire's compared to any other grid. This may be due to the minor variations in cell sizes within the random grid. The regular grids have a set number of directions fire can spread, to reach neighbouring polygons; they also struggle with following the irregular patterns in Earth's terrain as seen by the lake (Figure 3.10). Overall, there is no definitive best grid type. It really depends on what the user is trying to simulate and how detailed they are willing to have the simulation be in exchange for more computational time. Another approach would be to make adaptive grids which dynamically create additional polygons along the fire line and reduce the number of polygons in areas that have already burned. This may help perform more detailed simulations efficiently, but the creation and destruction of polygons may also negatively impact performance.

Comparing the IGS and ForeFire produced a table examining the computational efficiency and similarity of their outputs (Table 3.2). ForeFire produces the same results every time it is run. For this comparison the IGS used the focused grid which shows variability in grid layout, producing different results each time it is run. The IGS wildfires that ran for shorter durations tended to be computationally faster than the wildfires that ran for longer durations, relative to ForeFire. This may be caused by the increasing size of the simulated

wildfire as it burns, resulting in more ignited polygons that can spread fire to their neighbours therefore requiring additional computation per iteration. The IGS produces a larger area of false positives than that of false negatives when compared to ForeFire; this is due to the fire line selection process. As the IGS selects the fire line from the outer edges of boundary polygons, regardless of how far the fire has spread within that polygon to its surrounding edges. The threat score for similarity between the IGS and ForeFire increases with the length of simulation. This is due to the overall area of the wildfire, which tends to be mostly true positives, growing at a faster rate than the boundary of the wildfire where false positives and false negatives tend to occur.

Uncertainty of results could be measured by performing multiple simulations and drawing a heat map based on how many simulations predict an area will be burned [153]. IGS inputs could be varied based on the confidence of each input to account for any lack of accuracy in the inputs. This study could be completed as future work.

3.7 Conclusions

A novel software platform that allows for the change of grid type using simple parameterisation for wildfire modelling was developed. This allowed for grids to be compared within the same framework. The software included irregular grids constructed from Voronoi diagrams. The approach of using irregular grids to predict the spread of wildfires allows for efficient computing while retaining a reasonable level of similarity to existing methods defined by the threat score. Existing software such as ForeFire can produce more precise simulations as it models fires in continuous space by moving markers representing the fire line, instead of simulating the spread of fire on a fixed grid determined before running the simulation. The use of continually moving markers requires additional computational time during simulation. The spatial resolutions of both programs must be noted as ForeFire has a spatial resolution of $30m^2$ throughout, while the focused grid has a varying spatial resolution with a maximum resolution of $10m^2$. The IGS has been incorporated into DecaMap's Common Operational Picture software platform, an application that has potential to assist with wildfire management (Chapter 6) [43].

The IGS could be improved in future work. A better form of site placement than the focused grid could be developed to prevent the resolution of site placement decreasing at such a high rate at distances further from the wildfire source. A higher resolution could also be used to reduce the modulated boundary pattern produced by the cubic splines when converting the IGS output to a line. Porting the rest of the IGS from Python to a more computationally efficient programming language such as C++ would speed up the setup process for the IGS, making the overall software more efficient.

For the results ForeFire was treated as the ground truth as its continuous nature can be made to produce results of a higher resolution than that set in the IGS experiments. IGS and ForeFire used the same wildfire model. ForeFire could be considered an industry standard as it is commonly referenced in relevant literature [154] [155]. If a dataset containing a timeseries of wildfire burn data in the Wicklow Mountains, Ireland was released the next step would be to examine the performance of the IGS using this real data. Controlled burning may also provide a source of this data.

The IGS uses land cover maps are based on CORINE Land Cover which records data over a pan-European area [156]. The IGS could be expanded to areas outside of Europe using land cover maps and fuel data for the land cover describing these areas.

This study has shown that the focused grid is the best option for greater resolution near the wildfire source compared to other grid types. The regular grids are a better choice for more efficient computation. The decision of which grid type to choose should be made on a case-by-case basis by the user. The study also showed that the IGS runs approximately 34 times more quickly than ForeFire with a mean similarity of 0.8. While ForeFire tends to produce wildfire simulations at a greater spatial resolution than the IGS. If computational efficiency is a concern where multiple wildfires may need to be simulated at one location, the IGS may end up being an alternative option.

Chapter 4 A Method to Identify Wildfire Ignition Points and Propagation Durations from Burn Scars Using Genetic Algorithms

A critical research area regarding wildfire modelling often overlooked is the task of finding where a wildfire started and how long that wildfire burned. A literature review revealed that there are no automated methods with the goal of estimating the location of ignition points and the propagation duration of a wildfire from a burn scar. This chapter describes a novel method built in Python called the Wildfire Source Genetic Algorithm (WSGA). It can estimate the ignition points and the propagation duration of a wildfire, given the wildfire's burn scar, and environmental conditions used as input for a forward running wildfire simulator. Two sources of burn scar data were used as inputs for separate analysis. The first source was from a wildfire simulator which modelled strategically placed wildfires in the Wicklow Mountains, Ireland to test a wide range of scenarios [157]. Twenty generated wildfires from the wildfire simulator were used to validate the WSGA, as the location of the ignition points and propagation duration of each wildfire was known. The second source was real burn scars from the Wicklow Mountains, Ireland [103]. Given an input burn scar, the WSGA generates sets of potential ignition points and propagation durations, which are then simulated and compared to the input burn scar. These potential ignition points and propagation durations are iteratively improved using a genetic algorithm. A comparison is made between the ignition points and propagation durations of the simulation generated input burn scars and the best fitting WSGA seeded burn scars. This is done by evaluating two measures of error developed in this paper called the relative distance error and relative simulation duration error. The relative distance error measured error in the placement of ignition points. The WSGA had a relative distance error of 0 to 1.25 times the diameter of generated input burn scars. Lower errors were associated with larger wildfires. The WSGA had a relative simulation duration error of 0.0006 to 0.49 times the propagation duration of generated input burn scars. This study showed that it is possible to estimate the ignition points and propagation durations of wildfires from burn scars using genetic algorithms which could become a useful tool for fire investigators.

There is a description of resources and software used in Section 4.1, an explanation of the methodology including the genetic algorithm in Section 4.2, results in Section 4.3, followed by a discussion about the results in Section 4.4 and conclusions in Section 4.5.

4.1 Resources

The genetic algorithm (GA) in this chapter takes as input a burn scar (Figure 4.1). The GA will then generate a list of potential ignition points and propagation duration combinations. A forward running wildfire simulator is used to simulate the combinations of ignition points and propagation durations in the list producing a set of seeded burn scars. The set of seeded burn scars is then compared to the inputted burn scar. The best fitting burn scars in the set are selected, and their ignition points and propagation durations are modified before evaluating again. This step repeats for a user defined number of iterations, this is described in greater detail in Section 4.2. Afterwards the best fitting burn scar is outputted. The location of ignition points and propagation duration that form the outputted burn scar are then compared to the ignition points and propagation duration of the inputted burn scar if the data are available. Generated burn scars from a forward running wildfire simulator and real historical burn scars are both used.

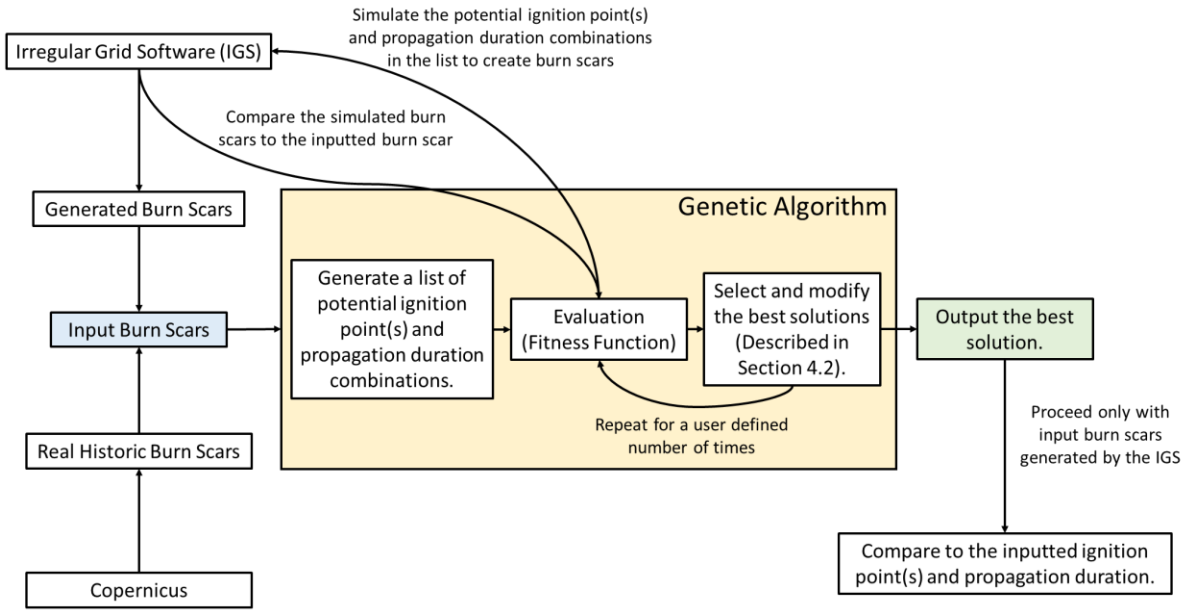

Figure 4.1: The steps used to estimate ignition points and propagation durations of inputted wildfire burn scars.

4.1.1 The Irregular Grid Software

The Irregular Grid Software (IGS) was selected as the forward running wildfire simulator; however alternative forward running wildfire simulators could be used (Chapter 3). The IGS ran the Rothermel model for simulations [46] [47]. The IGS takes as input elevation data, land cover data, land cover fuel data and wind data. The data is only used within the IGS. The elevation and land cover data were sourced from Copernicus [97]. The land cover map was developed using a random forest algorithm to classify the ground's surface into CORINE land cover classes [157] [101]. The combustion properties of each land cover type were sourced from another wildfire simulator called ForeFire [141]. The inputted weather conditions included wind speed and direction. To provide a focus on the inverse problem the wind was kept constant throughout all simulations. For the generated input burn scars, different wind speeds were selected for each burn scar to test the genetic algorithm. For the historical, real burn scar dataset, the wind data were sourced from the Irish state meteorological service called Met Éireann. Met Éireann have historic weather data which were used to approximate the wind conditions during the wildfires that created the burn scars [158].

The IGS was selected as the forward running wildfire software as it discretises (splits up) the area of interest using a grid containing a finite number of polygons. This allows the IGS to run multiple simulations in the same area faster than other existing software which run on continuous planes such as ForeFire [66]. This efficiency was important as in Section 4.2.2 the IGS is used to run many wildfire simulations seeded by the GA. The IGS positions a user defined number of sites (points) over the geographic region of interest [157]. These sites are then used to seed a Voronoi diagram. For this study 2,000 sites were positioned in random geographic positions within the region of interest to create a traditional random Voronoi diagram of polygons. The grid on which a particular input burn scar is presented on, and the GA trials its generated sets of ignition points and propagation durations contain the same polygonal arrangement. It is important to note that when the IGS discretises an area it gets the mean elevation and fuel properties from all pixels inside the polygon. Therefore, wildfires may propagate in polygons that contain non-ignitable fuel once these polygons also contain some ignitable fuel. An example of this can be seen in Section 4.3, Table 4.2 where one of the ignition points for wildfire 20 is located in Lough Tay, Wicklow, Ireland.

4.1.2 Generated Burn Scars

Twenty burn scars were strategically generated in the Wicklow Mountains, Ireland using the IGS, with a selected number of placed ignition points and propagation durations to test a variety of scenarios. The Wicklow Mountains, Ireland was chosen due to local knowledge, access to the region, and as it is commonly used by members of the Irish Fire Service for training exercises. The burn scars produced by simulating wildfire propagation from these ignition points for their respective propagation duration were used as input burn scars. An important point to note is that the IGS is also used later in Section 4.2.2 for all inputted burn scars, to evaluate GA generated ignition points and propagation durations. This means that the same weather conditions, terrain properties and forward running wildfire simulators (using the same wildfire propagation models) are used for generating input burn scars and evaluating burn scars seeded with data from the GA. This removes any error that could be due to weather data, terrain properties or the forward running wildfire simulator. However, it also means that the robustness of the genetic algorithm's ability to produce correct

results when presented with erroneous weather conditions, terrain properties or forward running wildfire simulators is not tested.

4.1.3 Real Historic Burn Scars

Three real historical burn scars from the Wicklow Mountains, Ireland were also used as input burn scars [103]. There was no existing data regarding the ignition points and propagation durations of these real burn scars. The real historic burn scars were represented by geocoded polygons. The IGS discretised a region of interest surrounding and including the burn scars into polygons. The GA estimated propagation durations and ignition points for these burn scars in Section 5. Although the real burn scars had no recorded ignition point or propagation duration data, the chromosome with the highest fitness score from the GA was simulated and visually compared to the real burn scar. The real burn scars were not used as a method of evaluating the GA but rather they show how this novel development could be useful in real world applications.

4.1.4 Genetic Algorithm Polygon Data

The region of interest surrounding and including the input burn scars consist of a set of contiguous polygons which are classified as vegetation (unburnt) and ignited (burnt) terrain. These categories are used in Section 4.2.2 during evaluation.

The GA in this chapter required a method to trial sets of generated potential ignition points and propagation durations using the IGS, so they could be compared to the input burn scar. A method was developed to encode the polygonal grid from the IGS into a bitstring to be interpreted by the GA (Figure 4.2). Each polygon in the IGS grid was labelled with an index ranging from 0 to the total number of polygons (minus one) on the grid. This index correlated to the position of a bit in the bitstring with the total number of polygons in the IGS grid being its length, where index 0 was the first bit in the bit string. Any polygons that contain an ignition point are called source polygons (their sites are treated as ignition points for calculations). Wildfire ignition points are points in space, however while using the IGS their resolution is limited to the polygons that encompass them. The use of polygons to

represent starting positions means that there is a potential loss of spatial precision as the spatial resolution can only be as small as the polygons. Every polygon was given a binary parameter of either being a source polygon or not, where there could be more than one source polygon. If the value in the bitstring was '1' then that polygon was a source polygon and if it was '0' then the polygon was not. Propagation durations (simulation durations) are stored as a positive integer. It is important to note that the mentioned data are a record of the source polygons and simulation durations for chromosomes in the GA, not the input burn scar. Multiple combinations of ignition points and propagation durations are generated in this format to be compared with the input burn scar and modified. This is discussed in Section 4.2. The IGS only required a simulation duration and the bitstring to seed a wildfire. Polygons from the IGS have other data associated with them such as their class (vegetation, burning or burnt) which are used for other applications [157] and are not relevant for seeding a wildfire.

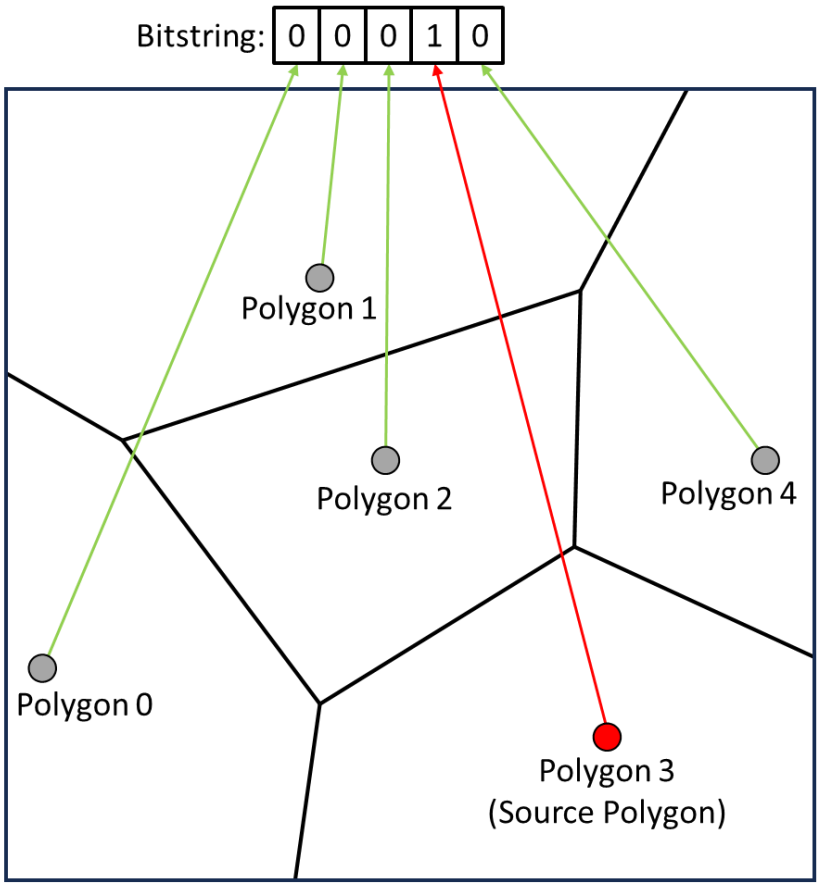

Figure 4.2: Example of the IGS polygons being converted to a bitstring by index. Polygons with grey sites are not wildfire source polygons while polygons with red sites are. In the bitstring the bits with a value of 1 represent polygons that are source polygons and bits with a value of 0 represent polygons that are not.

4.2 The Genetic Algorithm

The genetic algorithm presented called the WSGA (Wildfire Source Genetic Algorithm) estimates the source polygons and simulation duration of an input burn scar. It runs for a user defined number of generations (unless a solution with a perfect fitness score is found (Section 4.2.2)). The WSGA uses a chromosome description that consists of two parts. One part is a bitstring that represents the polygons in the IGS simulation (a binary coded polygon, source or not), and the other part is a positive integer value that represents the duration the wildfire should propagate from its ignition points (Figure 4.3). This allows for a WSGA chromosome to seed an IGS wildfire simulation as chromosomes hold information regarding source polygons and simulation duration. These information in these chromosomes can be modified by later steps in the WSGA. There is an upper limit set on the simulation duration. The user defines this value, and it should be chosen to be longer

than any likely fire duration but not so long as to increase the temporal limit unnecessarily. The WSGA has multiple variables like this that can be changed by the user before starting the WSGA. The values of user defined variables used in this study are displayed in Section 4.2.13, Table 4.1.

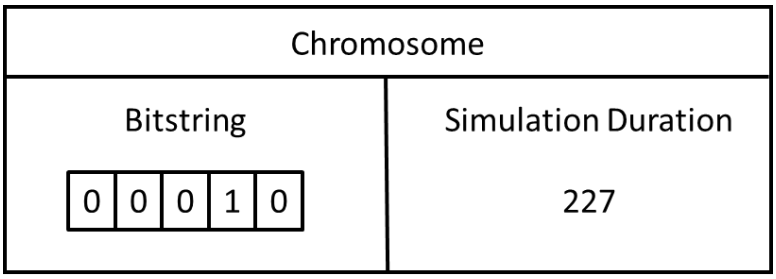

Figure 4.3: Example of a chromosome with a bitstring of genes that represent the state of polygons for the IGS simulation and a positive integer value that represents the simulation duration in seconds.

The WSGA follows most steps of traditional GAs, but some were modified, and completely new steps were introduced to help the GA specialise in the task of estimating wildfire ignition points and propagation durations (Figure 4.4). The remainder of this section explains each step in the GA.

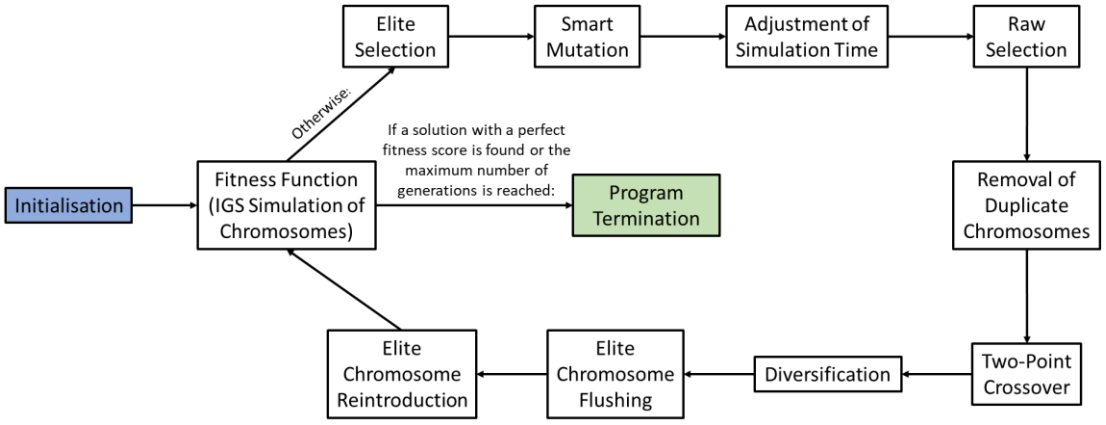

Figure 4.4: The steps required to estimate wildfire sources and simulation durations using the WSGA.

4.2.1 Initialisation

The WSGA starts by creating an initial list of chromosomes forming the first generation's population. The number of chromosomes in each generation's population is a user defined value. A larger number of chromosomes per generation increases diversity among

chromosomes but it also increases computational time for each generation. For each chromosome in the population a bitstring was generated with the same length as the number of polygons in the IGS grid with each bit representing a polygon in the IGS grid. The value of every bit in the bitstring was generated by drawing 1's and 0's (wildfire source or not) from a Bernoulli distribution independently [159]. All bits shared an equal independent user defined probability to have a value of '1', otherwise that bit would have a value of '0'. As this process ran for every bit, in the bitstring, there could be anywhere between zero to the number of bits in the bitstring with a value of '1'. This process defined where source polygons would exist on the grid. The source polygons can appear anywhere on the grid and were not constrained to just inside the input burn scar. There was also no requirement for a source polygon to be neighbouring any other source polygons. The simulation duration was drawn from a uniform distribution between the values of 1 second and the user defined maximum simulation duration. A minimum simulation time of 1 second was selected as the fire would not have a chance to propagate below this value. However, wildfires typically propagate for much longer than a second [92]. These two components formed the chromosomes in the initial population (some of these chromosome components are later modified by the other steps in the WSGA). This population was then evaluated using the fitness function, starting with the first generation.

4.2.2 Fitness Function (IGS Simulation of Chromosomes)

Every chromosome in the population list was simulated using the IGS where their source polygons were set by the bitstring. The wildfire then propagated from these source polygons for the simulation duration specified by the chromosome. All IGS simulations were run in C++ for increased computational efficiency, although Python could also be used. It is important to note for the generated burn scars, the IGS simulations in the fitness function used the same land cover maps, land cover fuel data, elevation maps, wind data and wildfire model as were initially used to generate these input burn scars. The fitness function can be described as a simple function that quantifies the fraction of simulated burn scar that match the input burn scar. This was done by comparing the burn status of polygons between the IGS outputted burn scar seeded by the WSGA and the input burn scar. For the fitness function, polygons could be in one of two classes, either vegetation or ignited (these classes

are only used to calculate a fitness score, they do not alter the bitstring of the chromosome in any way). Any polygons that could propagate the wildfire (polygons that contain ignition points or polygons that have had the wildfire propagate to their site) are in the ignited class while all other polygons are in the vegetation class. For most of the results in this study the two classes were equally weighted when calculating the fitness score, but there is an option to have different weights selected by the user. The optimal fitness score was calculated by checking every polygon in the input burn scar and summing up the weights associated with each polygon class as shown in (Eq. 4.1) and (Eq. 4.2). This provided a means of relative scoring for the chromosomes.

$$S(x) = \begin{cases} w_I, & if\ x \in I \\ w_V, & if\ x \in V \end{cases},$$

(Eq. 4.1)

$$Q = \sum_{i=0}^{n_p} S(O_i),$$

(Eq. 4.2)

where x is a polygon, $S(x)$ is a function that checks the class of a polygon, if $x \in I$ then polygon x is ignited and the wildfire has propagated to its site, if $x \in V$ then polygon x is still vegetation and the wildfire has not propagated to its site, $w_I \in \mathbb{R}_{>0}$ is the weight for ignited polygons, $w_V \in \mathbb{R}_{>0}$ is the weight for vegetation polygons, $Q > 0$ is the optimal fitness score calculated by getting the number of ignited polygons times their weight, summed with the number of vegetation polygons times their weight, $n_p > 0$ is the total number of polygons, i is the index of a polygon and O_i is the polygon at index i of the input burn scar.

The similarity between input burn scar and the WSGA chromosome seeded burn scar was found by comparing the classes of polygons of the same index in each burn scar. If the class of a selected polygon in the chromosome's burn scar and the input burn scar were the same then the fitness score increased based on that class's weight, otherwise the score did not increase (Figure 4.5) (Eq. 4.3) (Eq. 4.4). Every polygon within the chromosome was compared to the same polygon in the grid from the input burn scar.

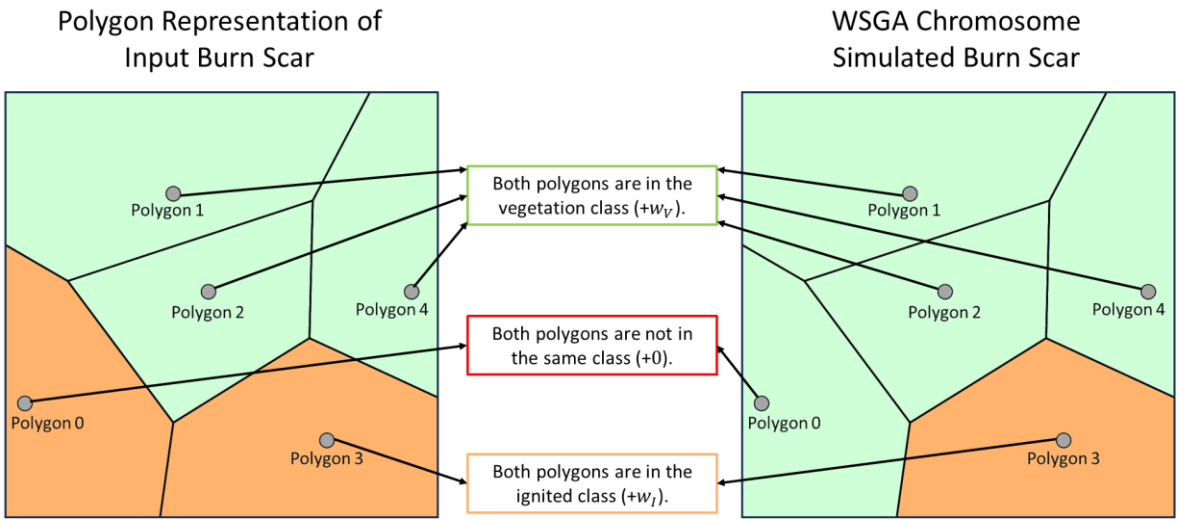

Figure 4.5: An example of the comparison between polygons of the input burn scar and a simulated chromosome burn scar where green polygons are in the vegetation class and orange polygons are in the ignition class. If polygons of the same index are vegetation, then a weight w_V is added to the calculation of the fitness score, if they are ignited then a weight w_I is added and if they are not in the same class then nothing is added.

$$P(x,y) = \begin{cases} w_I, & if\ x,y \in I \\ w_V, & if\ x,y \in V \\ 0, & if\ x \in I; y \in V \\ 0, & if\ x \in V; y \in I \end{cases},$$

(Eq. 4.3)

$$F = \frac{\sum_{i=0}^{n_p} P(C_i, O_i)}{Q},$$

(Eq. 4.4)

where y is a polygon, $P(x,y)$ is a function that checks the classes of polygons x and y from the input and WSGA chromosome simulated burn scars, F is the fitness score of the chromosome relative to the optimal fitness score, and C_i is the polygon at index i of the simulated chromosome burn scar.

This process produced a fitness score which was used to rank chromosomes in terms of similarity to the input burn scar. If the fitness score was 0 then the chromosome seeded burn scar had no polygons of the same index in the same class as the input burn scar. If the fitness score was 1, the chromosome seeded burn scar was a perfect match with the input burn scar for each polygon. If there was a perfect match or the maximum number of

generations was reached, the WSGA terminated. Otherwise, the WSGA continued with elite selection.

4.2.3 Elite Selection

Elite selection involves checking the fitness scores of the chromosomes after every generation. It is part of a process called elitism; a technique commonly used in genetic algorithms that preserves the best chromosomes [160]. Without the use of elitism, the WSGA continued to provide good solutions occasionally, but some of the best chromosomes determined by the fitness function were lost in previous generations. Preserving the best chromosomes between generations allowed the WSGA to improve these specific chromosomes iteratively and output the best solution at the end. For elitism in the WSGA a new list called an elite list was created to separately store elite chromosomes; it had a user defined length. If a new chromosome had a fitness score that is greater than any of the chromosomes already in the elite list, or if there was empty space in the elite list, then a copy of that new chromosome from the population was placed in the elite list. If there was no empty space on the elite list, the existing chromosome with the lowest fitness score already in the elite list was removed.

4.2.4 Smart Mutation

Smart Mutation was used to make minor adjustments to the source polygons of a chromosome. It replaced the traditional form of bit-flipping mutation used in many genetic algorithms, where every gene has a random chance of changing value [161]. Smart mutation only affected chromosomes that had a fitness score above a certain user defined threshold. The bitstrings of chromosomes above this threshold were checked. Once a bit with a value of '1' was found, the algorithm used the index of that bit in the bitstring to find the polygon of the same index and looped through all its neighbouring polygons. While looping through all neighbouring polygons there was a user defined chance for the source polygon to mutate to a non-source polygon and for a neighbouring polygon to then mutate into a source polygon. Essentially swapping the bit values of these two neighbouring polygons within the bitstring. Once this happened the process halted for that original

polygon (bit) and smart mutation continued to search the remainder of the bitstring for bits with a value of '1'. This process allowed neighbouring polygons to swap the status of source polygon between each other to improve the probability of obtaining a higher fitness score in the next generation (Figure 4.6).

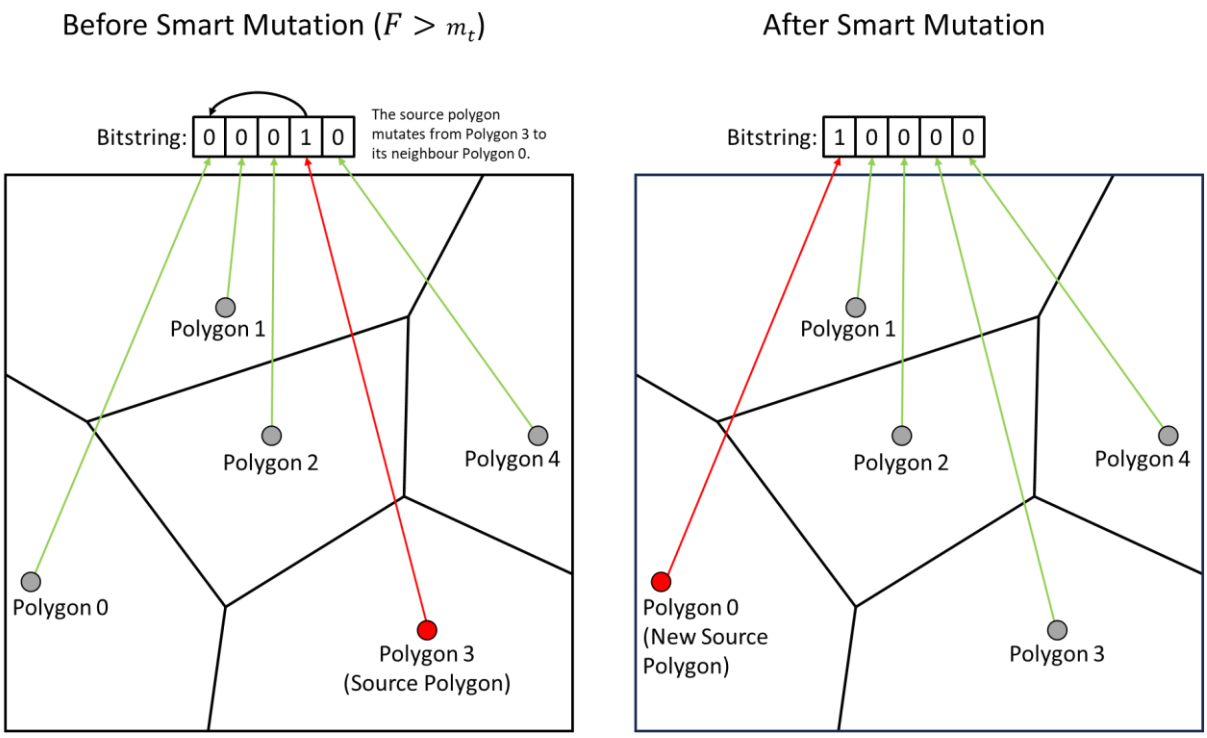

Figure 4.6: Example of smart mutation swapping which polygon is a source polygon. Smart mutation can only perform this action between neighbouring polygons. Polygons with grey sites are not wildfire source polygons while polygons with red sites are. In the bitstring the bits with a value of 1 represent polygons that are source polygons and bits with a value of 0 represent polygons that are not. F is the fitness score of the chromosome and m_t is the user defined fitness score threshold for smart mutation to occur.

4.2.5 Adjustment of Simulation Duration

The simulation durations of all chromosomes were updated every generation, to try and increase the fitness score of the chromosomes in future generations. The area of the chromosome simulated burn scar was compared to the area of the input burn scar (unlike the fitness function, this was only a comparison of areas to gauge how much the simulation duration of the chromosome should be adjusted). Area was chosen as a reliable comparison as the burn scars of wildfires tend to increase in area over time when the fire is still burning [162]. If the area of the burn scar from a simulated chromosome was greater than the area of the input burn scar, then that chromosome's simulation duration decreased for the next

generation. If the area of the burn scar from a simulated chromosome was smaller than the area of the input burn scar, the chromosome's simulation duration increased. If the area of the chromosome burn scar and the input burn scar were the same, then the chromosome's simulation duration did not change. The adjustment of simulation duration is applied to the chromosomes, therefore the remaining steps in the WSGA involving the chromosomes use the updated simulation duration affecting the next generation. Later steps in the WSGA such as two-point crossover and diversification provided an alternate mechanism for varying the duration. The updated simulation duration for a particular chromosome in seconds was found using:

$$r_{C_{g+1}} = \text{round}\left(r_{C_g} w_t \left(\frac{A_O - A_C}{A_O} + \frac{A_O - A_C}{A_C}\right)\right),$$

(Eq. 4.5)

where $r_{C_{g+1}} \geq 1$ is the new simulation duration for the chromosome (s), $r_{C_g} \geq 1$ is the previous simulation duration for that chromosome (s), $w_t > 0$ is a weight which alters the amount the simulation duration will change, $A_O > 0$ is the area of the of the input burn scar (m^2) and $A_C > 0$ is the area of the burn scar from WSGA chromosome seeded wildfire (m^2).

In equation (Eq. 4.5) the term $\frac{A_O - A_C}{A_O}$ decreases linearly as A_C increases. This is perfect for influencing the rate at which $r_{C_{g+1}}$ should decrease when $A_C > A_O$. However, if $A_C < A_O$, and A_C approaches the value of 0, the term $\frac{A_O - A_C}{A_O}$ approaches the value of 1 which does not sufficiently influence the rate at which $r_{C_{g+1}}$ should increase. Therefore, the term $\frac{A_O - A_C}{A_C}$ was added to the equation which exponentially increases when $A_C < A_O$, and A_C approaches the value of 0. The two terms added together, exponentially increase the value of $r_{C_{g+1}}$ when $A_C < A_O$, and decrease the value of $r_{C_{g+1}}$ in an approximately linear fashion when $A_C > A_O$ (Figure 4.7). The two terms can also be simplified as seen in (Eq. 4.6).

$$\frac{A_O}{A_C} - \frac{A_C}{A_O} = \left(\frac{A_O - A_C}{A_O} + \frac{A_O - A_C}{A_C}\right)$$

(Eq. 4.6)

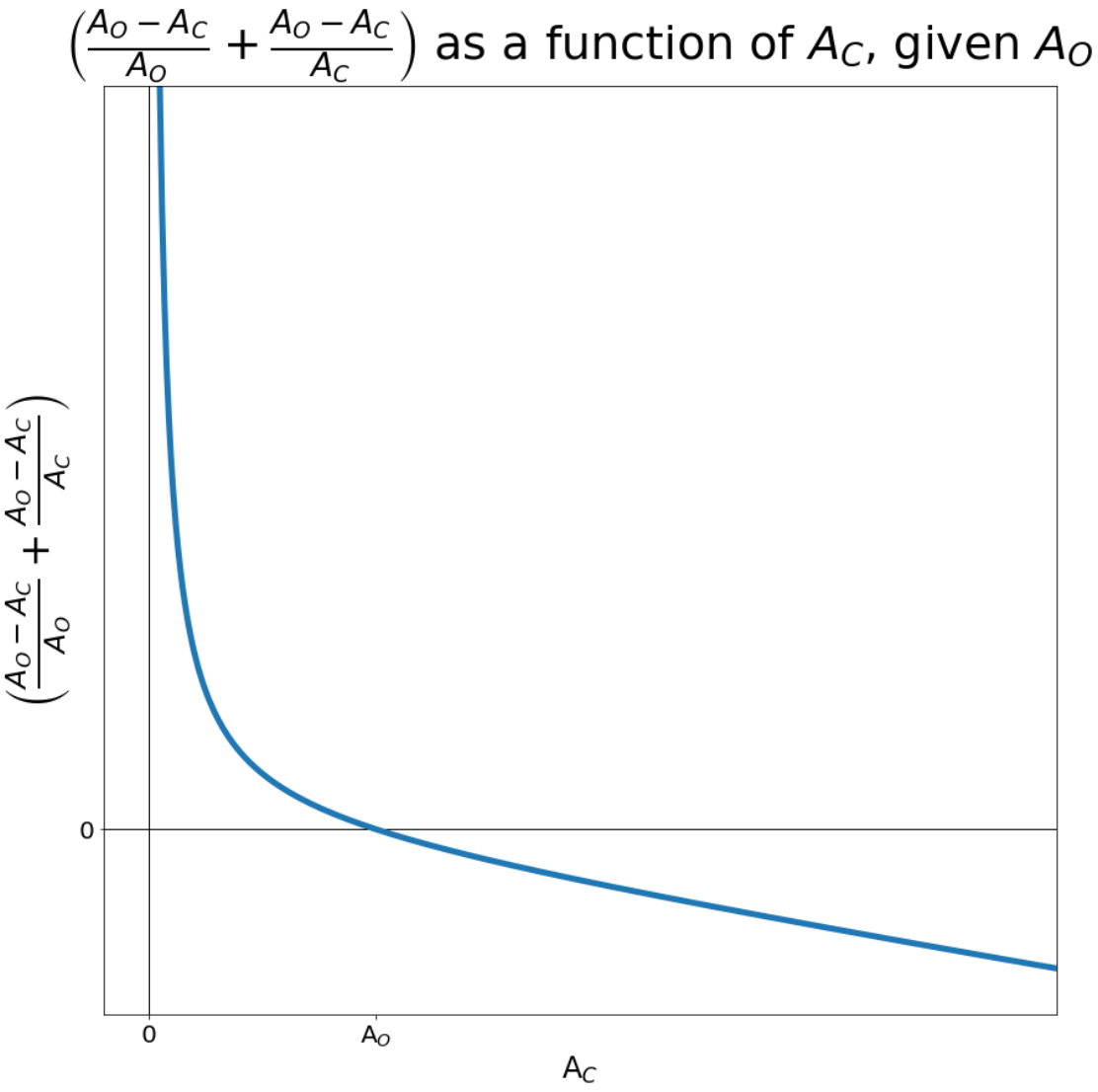

Figure 4.7: Graph of $\left(\frac{A_O - A_C}{A_O} + \frac{A_O - A_C}{A_C}\right)$ as a function of A_C, given A_O.

4.2.6 Raw Selection

Raw Selection was used to select chromosomes based on their fitness score. A user defined number of chromosomes were chosen using raw selection. The chromosomes were selected in descending order of fitness score from the fitness function.

4.2.7 Removal of Duplicate Chromosomes

The set of chromosomes compiled using raw selection were compared. If any chromosome had the same fitness score and bitstring as another chromosome, then one of them was replaced with a newly generated random chromosome. The creation of a new chromosome was identical to the process of generating chromosomes in the initialisation step of the WSGA. This increased diversity among the list of chromosomes for the following steps in the WSGA.

4.2.8 Two-Point Crossover

Two-point crossover was used to combine the bitstrings within all chromosomes from the previous step with one another to produce new chromosomes using the combined bitstrings, analogous with meiosis [163]. As raw selection extracted chromosomes with high fitness scores, combining bitstrings from these existing chromosomes has a chance to generate chromosomes with even higher fitness scores. Two-point crossover also increases genetic diversity in the newly produced chromosomes. The combining process occurred with two chromosomes from the previous step in the WSGA at a time which generated two new chromosomes each time. For example, given two chromosomes A and B, two-point crossover works as follows. To begin, two positions in the bitstrings known as indices are randomly selected, these indices are the same for both chromosomes A and B (for k-point crossover, k number of indices would be selected). The segments of the bitstrings between the start, crossover indices and end of the bitstrings are split up into smaller bitstrings called substrings for both chromosomes A and B. New chromosomes C and D are then created by swapping every odd-positioned substring between chromosomes A and B (Figure 4.8). For this algorithm the crossover indices could be randomly placed before the first or after the last bit in the bitstring which reduced the number of crossover indices and number of substrings which would be swapped. The simulation durations for chromosomes C and D are the rounded mean simulation duration from chromosomes A and B (Eq. 4.7).

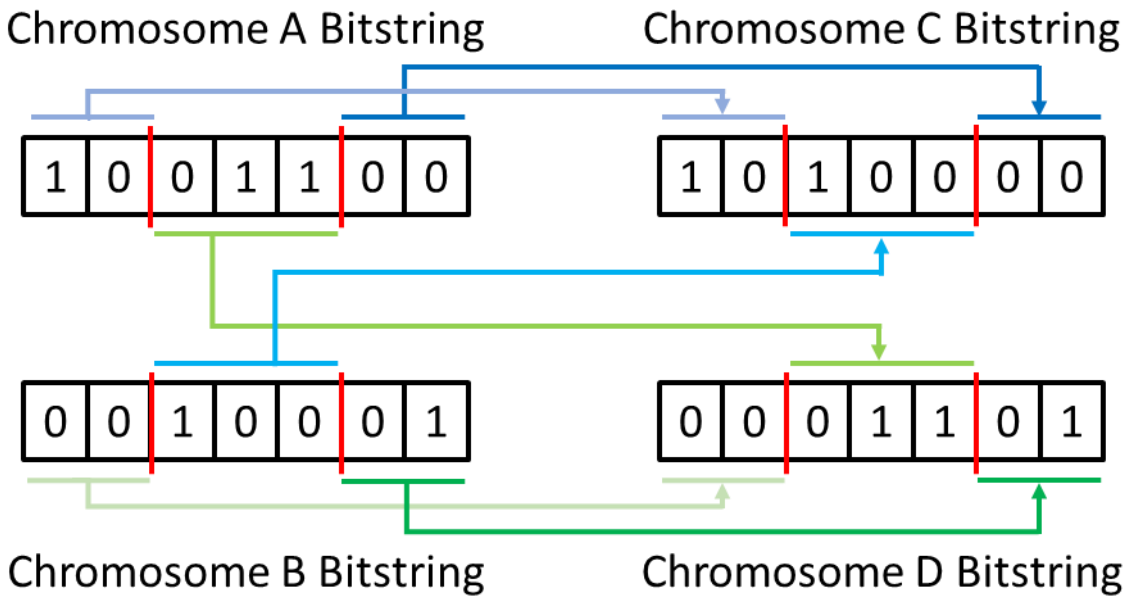

Figure 4.8: An example of two-point crossover between bitstrings from chromosomes A and B, to create bitstrings for chromosomes C and D. The indices selected for this example of two-point crossover were 2 and 5 and are represented by red lines. The parts of the bitstrings from chromosome A and B that make up the bitstring for chromosome C, are blue while the parts that make up the bitstring for chromosome D, are green.

$$r_C = r_D = \text{round}\left(\frac{r_A + r_B}{2}\right), \quad \text{(Eq. 4.7)}$$

where r_C, r_D, r_A and r_B are the simulation durations for chromosomes C, D, A and B respectively (s).

Running this process on the set of chromosomes from the previous step can produce many new chromosomes as every chromosome in the set goes through the process of two-point crossover with each other. The total number of new chromosomes that will be produced can be found using (Eq. 4.8).

$$n_c = (n_r - 1)^2 + (n_r - 1), \quad \text{(Eq. 4.8)}$$

where n_c is the total number of new chromosomes produced from crossover and $n_r \geq 2$ is the number of chromosomes selected during raw selection.

These new chromosomes formed a new list. If the number of new chromosomes produced was above the user defined number of chromosomes for each generations' population, then chromosomes were randomly removed from the new list until its length was the same as the user defined value. If the number of new chromosomes produced was below the user defined number of chromosomes in each generations' population, then chromosomes were generated (like Section 4.2.1) until the number of new chromosomes in the new list was equivalent to the user defined number of chromosomes per generation.

4.2.9 Diversification

Diversification is the process of adding unique chromosomes to help prevent the WSGA from getting stuck at a local optimum [164]. In this study diversification was performed by adding new original chromosomes (generated like the initialisation step) to the population every generation. Without diversification, populations in the WSGA tended to converge to one (typically sub-optimal) solution. Diversification solved this potential challenge by generating a pre-selected number of new chromosomes independently from those generated by two-point crossover. The newly generated chromosomes randomly replaced existing chromosomes in the list created from two-point crossover, introducing new and random chromosomes into the population.

4.2.10 Elite Chromosome Flushing

Elite chromosome flushing was developed to increase diversity when the WSGA had stopped developing better solutions. If a user defined number of generations passed without any new chromosomes entering the elite list, the entire population was removed and replaced with newly generated chromosomes (the elite list remains unchanged). As the number of generations in the WSGA increases, elite chromosome flushing occurs more often as the existing chromosomes in the elite list tend to have high fitness scores, making it hard for new chromosomes to enter the elite list. Due to Section 4.2.11, elite chromosomes remained in the population.

4.2.11 Elite Chromosome Reintroduction

Elite chromosome reintroduction was the final step in every non-terminating generation. It replaced random chromosomes in the list with the set of chromosomes stored in the elite list. This process preserved the best chromosomes throughout every generation. After this step the new generation is defined and the WSGA looped back to run the fitness function on the new generation and continued the process.

4.2.12 Program Termination

Program termination occurred when the WSGA reached the user defined maximum number of generations or had a chromosome with a perfect fitness score. The WSGA extracted the best chromosome by fitness score from the elite list and outputted it. The chromosomes contained the bitstring and simulation duration that were estimated by the WSGA to be the best match for the input burn scar.

4.2.13 Generating Results

In Section 4.3 the WSGA is compared with the set of input burn scars. For these tests the following user defined values presented in Table 4.1 were used. For each of the simulations performed using the IGS, the time step was 1% of the simulation duration, with a minimum time step of 1 second.

Term	Variable	Value
Number of polygons in IGS simulations (length of bitstrings)	n_p	2,000
Maximum simulation duration (seconds)	n_t	1,000,000
Maximum number of generations	n_g	1,000
Number of chromosomes in each generation	n_q	$\text{round}\left(\frac{n_p}{200}\right)$
Probability to set the value of any bit in a chromosome bitstring to '1' when creating new chromosomes (probability to set a source polygon)	s	$\frac{1}{n_q}$
Fitness function weight for ignited polygons	w_I	0.5
Fitness function weight for vegetation polygons	w_V	0.5
Elite chromosome list length	α	$\text{round}\left(\frac{n_q}{5}\right)$
Adjustment of simulation duration weight	w_t	0.005
Number of chromosomes selected during raw selection	n_r	$\text{round}\left(\frac{n_q}{2}\right)$
Fitness score threshold for smart mutation to occur	m_t	0.85
Probability for smart mutation to occur to each neighbour of an affected polygon	m_p	0.25
Number of chromosomes generated in diversification	n_d	$\text{round}\left(\frac{n_q}{10}\right)$
Number of generations without a newly created elite chromosome before elite chromosome flushing occurs	n_α	30

Table 4.1: Explanation of variables used in the WSGA accompanied by the values that were used to generate Section 4.3.

To verify the WSGA, twenty IGS generated input burn scars were used. These burn scars were created by strategically placing ignition points and setting various simulation durations to generate a wide variety of input burn scars. The WSGA was then used to try and estimate source polygons and the simulation duration for each of these input burn scars. When a simulation had multiple ignition points, all ignition points began propagating simultaneously once the simulation started, and their respective wildfire fronts propagated for the entire simulation duration.

The precision of the source polygon positions estimated by the WSGA output were compared to the position of source polygons used by the IGS to generate the input burn scar. Their simulation durations were also compared. The error of WSGA source polygon placement was found by measuring the proximity of the WSGA estimated source polygons to the source polygons of the input burn scar. The distance between each of the input burn scar source polygons and their nearest WSGA source polygon was summed along with the

distance between each of the WSGA source polygons and their nearest input burn scar source polygon. The sum of distances between the WSGA and input burn scar source polygons was chosen as each individual distance contributed to the overall error. Therefore, using this method, errors in the number of source polygons that the WSGA estimated contributed to the overall error. Other methods such as finding the mean distances between WSGA and input burn scar source polygons would not capture this. In the case of duplicate distances (where a source polygon from the WSGA and input burn scar are both each other's nearest counterpart), only one distance was counted in the sum. This ensured a measurement was made that included the minimum unique distances between all WSGA source polygons and all their input burn scar counterparts and vice versa. Duplicate distances tended to occur when the WSGA estimated the same amount of source polygons in similar locations as to the input burn scar. The distance between polygons was found by getting the distance between the polygons' sites. The sum of these distances was then made relative to the mean diameter of the input burn scar, estimated from its area (Eq. 4.9).

$$u = 2\sqrt{\frac{A_O}{\pi}},$$

(Eq. 4.9)

where A_O is the area of the input burn scar (m^2) and u is the mean diameter for a wildfire with area A_O (m).

This allows a relative distance error between input burn scar and WSGA source polygons to be calculated using the sum of unique distances and the mean diameter of the wildfire (Eq. 4.10).

$$\varepsilon_s = \frac{D}{un_o},$$

(Eq. 4.10)

where D is the sum of unique distances from the input burn scar source polygons to WSGA estimated source polygons and from the WSGA estimated source polygons to the input burn scar source polygons (m), ε_s is the relative distance error between the input burn scar source polygons and the source polygons estimated by the WSGA and n_o is the number of source polygons that formed the input burn scar.

The relative distance error is dimensionless as it is relative to the area of the input burn scar. For example, if an input burn scar had a mean diameter of 100 metres and the WSGA had a relative distance error of 0.1, then the cumulative distance error between source polygons would be 10 metres (if the input burn scar had the same number of source polygons as the WSGA estimated).

The relative simulation duration error was also dimensionless. It was found by getting the difference between the input burn scar simulation duration and the WSGA outputted simulation duration, then dividing by the input burn scar simulation duration. This made the value relative to the input burn scar simulation duration (Eq. 4.11).

$$\varepsilon_t = \frac{|r_O - r_\alpha|}{r_O},$$

(Eq. 4.11)

where ε_t is the relative simulation duration error between the wildfire that caused the input burn scar and the wildfire estimated by the WSGA, r_α is the simulation duration of the WSGA chromosome (s) and r_O is the actual simulation duration for the input burn scar (s).

It took the WSGA a mean time of 3 hours to estimate the source polygons and simulation duration of all IGS generated burn scars using an Intel® Core™ i7-10870H Processor @ 2.20GHz. The process of simulating the wildfires represented by chromosomes could be completed in parallel which is an avenue for potential future work. This could greatly speed up the process. Computing could also be done on a Graphics Processing Unit (GPU) which could reduce computing time even further.

4.3 Results

Twenty IGS generated burns scars were used to validate the WSGA. They were chosen to have a varied set of wind speeds, simulation durations and number of ignition points (Table 4.2). This was to test the versatility of the WSGA. A comparison of the polygonal input burn scar and the WSGA seeded burn scars were recorded along with the fitness score of the best chromosome in the WSGA for each generation (Figure 4.9). A complete set of figures can be found in Section A.2 Figures for all WSGA Results.

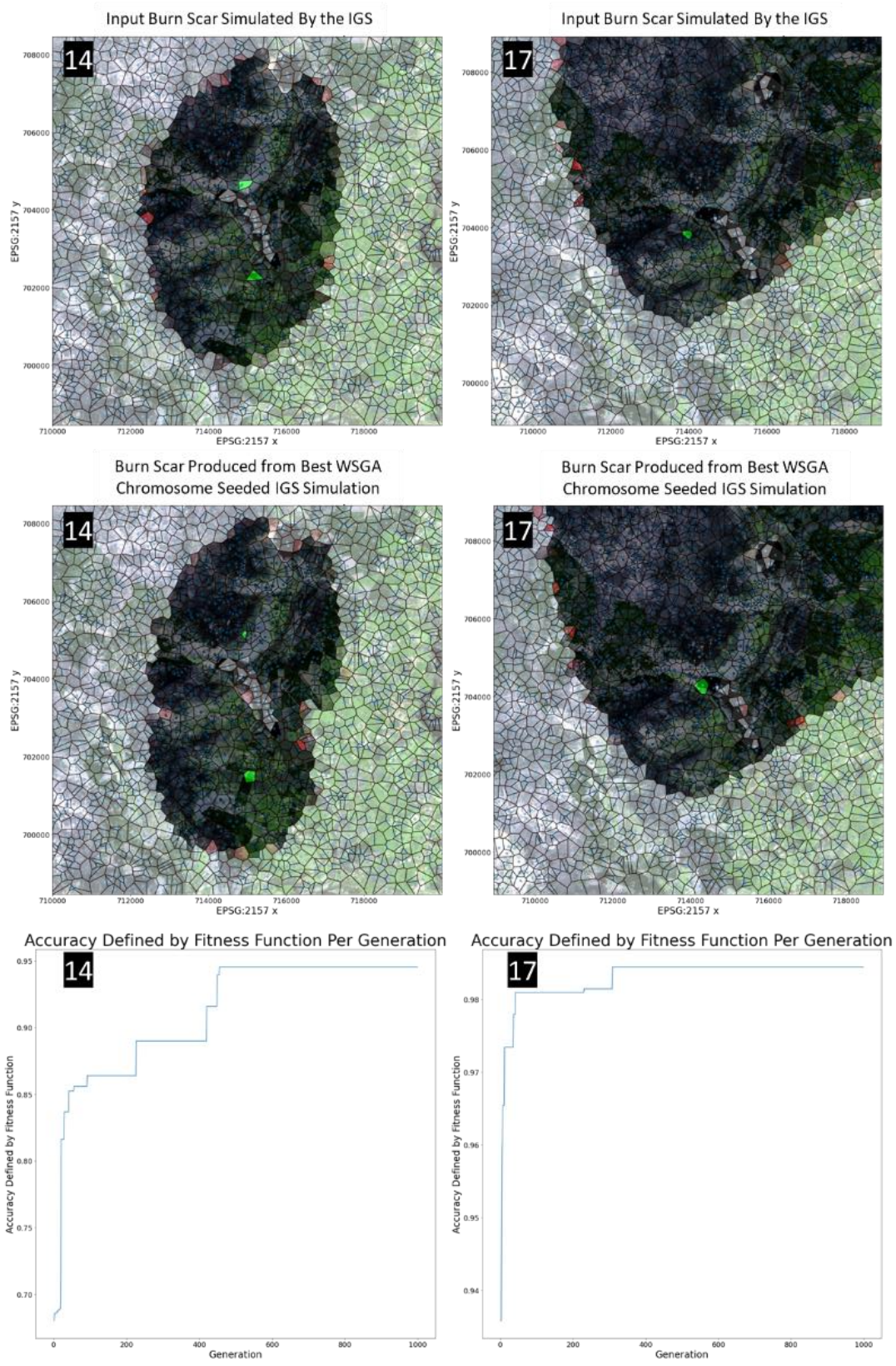

Figure 4.9: An example of the results obtained using the WSGA. The top row shows the input burn scars of Wildfire 14 and Wildfire 17 respectively, simulated using the IGS. The middle row shows the burn scars created by simulating the wildfires from the best chromosome produced by the WSGA using the IGS. The condition of the polygons is represented by their colour, not burnt (transparent), wildfire source (green), burning (red) and burnt (black) based on the mass of the polygon in that state, with a satellite image of the terrain in the background (Copernicus 2022). The visual outputs are cropped to show source polygons, but the simulations may continue using the grid outside of this cropped zone. The bottom row shows the fitness score of the best chromosome at every generation of the WSGA for these wildfires.

Wildfire	Wind (Zonal, Meridian) (m/s)	Fitness Score	Generations Required to Produce Solution	Input Burn Scar Source Polygon Locations (x, y) (EPSG:2157)	WSGA Estimated Source Polygon Locations (x, y) (EPSG:2157)	Relative Distance Error Between Source Polygons	Input Simulation Duration (s)	WSGA Simulation Duration (s)	Relative Simulation Duration Error	WSGA Computational Run Time (s)
1	(0,0)	0.97	1,000	(715271, 702304)	(715524, 702442) (715382, 702057)	0.09	100,000	95,237	0.05	11,605
2	(0,0)	0.97	1,000	(713365, 705054)	(713061, 705883) (714440, 705597) (712959, 704975) (713993, 704561) (712929, 704404)	0.56	100,000	79,738	0.2	12,277
3	(0,0)	0.94	1,000	(715676, 706553)	(715058, 706859)	0.1	100,000	97,706	0.02	11,867
4	(0,0)	0.95	1,000	(710850, 708042)	(710759, 708466)	0.06	100,000	101,910	0.02	12,332
5	(0,0)	0.98	1,000	(713507, 708387)	(713192, 709580) (714328, 709126) (712686, 708643) (712746, 708338) (713993, 707785)	1.26	50,000	25,552	0.49	11,627
6	(0,0)	0.97	1,000	(712615, 708870)	(712361, 710102) (712239, 709521) (713365, 709373) (712665, 709225)	1.35	50,000	22,839	0.24	10,430
7	(0,0)	0.99	1,000	(715829, 708259)	(715585, 708732) (715879, 708269) (715656, 707756)	0.42	30,000	16,456	0.45	11,532
8	(0,0)	0.98	1,000	(713547, 711187)	(713547, 711187)	0	75,000	70,331	0.06	11,893
9	(0,0)	0.95	1,000	(714358, 702047)	(713892, 702185)	0.07	110,000	109,932	0.0006	11,216
10	(0,0)	0.96	1,000	(715170, 702313)	(713760, 703132) (716335, 702570) (715251, 701604) (714470, 701170)	0.64	125,000	81,360	0.35	11,899
11	(0,0)	0.91	1,000	(704068, 703516)	(703479, 705242) (706024, 703999) (703043, 702767)	0.47	130,000	93,987	0.28	11,888
12	(0,0)	1	531	(713497, 707194)	(713497, 707194)	0	150,000	149,886	0.0008	7,207
13	(0,0)	0.93	1,000	(715210, 702264) (714287, 704975) (711692, 706730)	(711824, 707795) (711591, 705567) (715139, 705133) (715139, 701574)	0.09	100,000	83,724	0.16	12,996
14	(0,0)	0.95	1,000	(715271, 702264) (714855, 704650)	(714946, 705133) (715078, 701535)	0.1	75,000	74,052	0.01	11,565
15	(0,0)	0.97	1,000	(715180, 702530) (715849, 702935) (713770, 708111)	(714075, 708505) (715595, 702905)	0.14	30,000	42,058	0.4	11,117
16	(0,0)	0.98	1,000	(714734, 704699) (715423, 707657)	(715139, 707776) (714673, 704857)	0.05	50,000	53,694	0.07	11,449
17	(1, 3)	0.98	1,000	(713973, 703763)	(714308, 704295)	0.04	50,000	58,962	0.18	13,143
18	(1, -3)	0.99	1,000	(712665, 715703)	(712665, 715703)	0	30,000	28,616	0.05	11,564
19	(-0.5, - 0.4)	0.99	1,000	(713892, 707480)	(714723, 708130) (713507, 708091) (713902, 707056)	1.11	20,000	13,738	0.31	11,497
20	(-1, 0.1)	0.97	1,000	(712706, 715762) (716092, 707361)	(710120, 716432) (712310, 715180) (715342, 707815) (716366, 707480)	0.14	50,000	33,102	0.34	13,181
Mean		**0.97**				**0.33**			**0.18**	**11,614**
Standard Deviation		**0.02**				**0.44**			**0.16**	**1,241**
Standard Error		**0.01**				**0.1**			**0.04**	**277**

Table 4.2: Comparison of the relative distance error and relative simulation duration error between source polygons for twenty inputted wildfires generated using the IGS and the same wildfires simulated with properties seeded by the WSGA.

Histograms of the relative distance errors and relative simulation duration errors from Table 4.2 were generated to visualise the results (Figure 4.10).

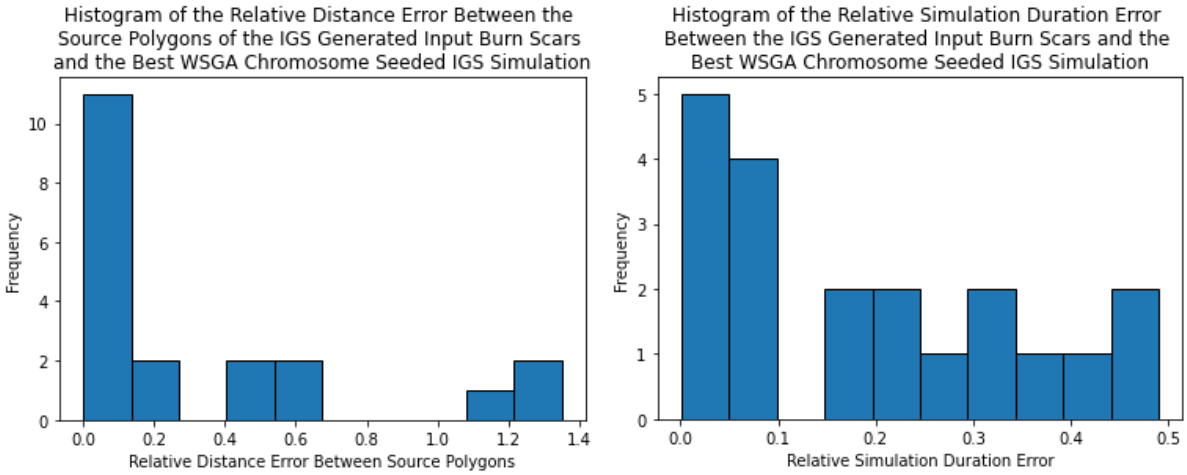

Figure 4.10: Histograms with values split into 10 bins of the relative distance error between source polygons and relative simulation duration error between the 20 IGS generated burn scars and WSGA seeded burn scars from Table 4.2.

The WSGA was also run on the three real burn scars covering areas of the Wicklow Mountains, Ireland. These burn scars were sourced from Copernicus [103], where they had the IDs 18433, 15164, and 284. For these simulations the fitness function weights were changed to $w_I = 0.75$ and $w_V = 0.25$, as the WSGA tended to overestimate the number of source polygons using the values in Table 4.1. Wind data were sourced using Met Éireann's historic dataset of wind speeds to find the daily mean Zonal (West to East) and Meridian (South to North) wind speed at the Casement weather station for the date listed by Copernicus for each burn scar. The wind speeds were as follows: 1.08m/s zonal and 1.87m/s meridian for burn scar 18433, −3.04m/s zonal and −0.54m/s meridian for burn scar 15164 and −3.34m/s zonal and 0m/s meridian for burn scar 284 [158]. Due to the real burn scars having no accompanying information regarding their ignition points or propagation times, it was not possible to calculate relative distance error or relative simulation duration error for these real burn scars. Running the WSGA only serves to estimate possible ignition points for these burn scars. It demonstrates the potential utility of the method (Figure 4.11).

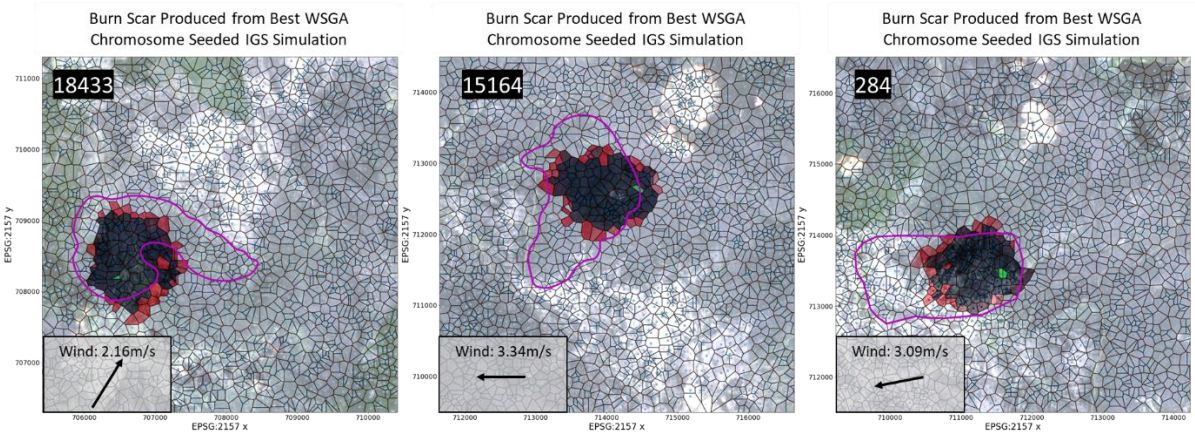

Figure 4.11: Burn scars of Wildfires 18433, 15164 and 284 (purple outline). The WSGA estimated the source polygons and simulation durations for these burn scars. The IGS then simulated the wildfires on a polygonal grid using this output from the WSGA. The condition of the polygons is represented by their colour, not burnt (transparent), wildfire source (green), burning (red) and burnt (black) based on the mass of the polygon in that state, with a satellite image of the terrain in the background (Copernicus 2022). The WSGA estimated simulation duration for wildfire 18433 was 11,074 seconds, wildfire 15164 was 11,153 seconds and wildfire 284 was 9,514 seconds.

In some cases, the WSGA seeded IGS simulated burn scars did not fit well with the real burn scars. The burn scar of wildfire 18433 was most likely shaped by the landscape in the region. The land cover data used by the IGS has a resolution of $10m^2$. Therefore, small rivers near wildfire 18433 that may have affected the actual wildfire may not influence the wildfire simulation (Figure 4.12) [165]. A finer polygonal grid and the manual inclusion of river data would likely produce a better fit. The wind data inputted into the WSGA for these real burn scars were recorded at the nearest weather station Casement, Co. Dublin, Ireland which is over 15km away from the real burn scars. The burn scars are also located at an altitude over 200m higher than the weather station. A weather station located on the Wicklow Mountains would provide more accurate data. Both wildfires 15164 and 284 may have occurred in different wind conditions. The WSGA was run again on burn scars 15164 and 284 but this time the zonal and meridian wind speeds were manually set to 0.1m/s and 3m/s respectively for wildfire 15164, and −6.5m/s and 0m/s respectively for wildfire 284. These parameters produced better results. This demonstrated the importance of high-quality wind data. The ignition point was estimated to be on the Southern section of burn scar 15164 and the Eastern section of the burn scar 284 (Figure 4.12). Other factors such as changes in land cover since the wildfire occurred could also influence the wildfire simulation. Temporal variations in the wind from real historic wind data could not be included in these simulations as the exact time these wildfires occurred was unknown.

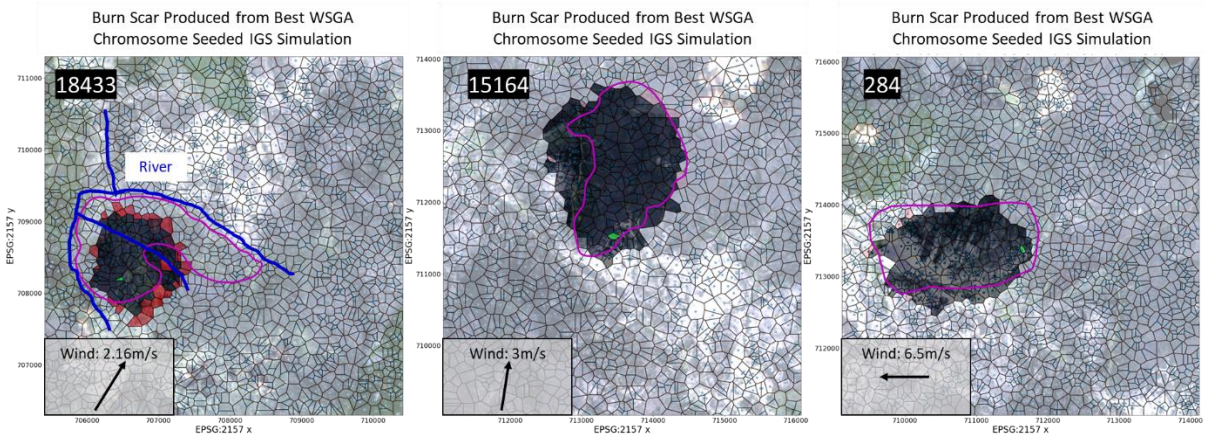

Figure 4.12: Burn scars of Wildfires 18433, 15164 and 284 (purple outline). Wildfire 18433 has rivers superimposed which may have affected the real wildfire but not the simulation. The WSGA estimated the source polygons and simulation durations of Wildfires 15164 and 18433 with new wind conditions. The IGS then simulated the wildfires on a polygonal grid using this output from the WSGA. The condition of the polygons is represented by their colour, not burnt (transparent), wildfire source (green), burning (red) and burnt (black) based on the mass of the polygon in that state, with a satellite image of the terrain in the background (Copernicus 2022). The WSGA estimated propagation duration for wildfire 18433 was 11,074 seconds, wildfire 15164 was 10,319 seconds and wildfire 284 was 4,152 seconds.

4.4 Discussion

The results show that WSGA was able to estimate the source polygons and simulation duration of IGS generated input burn scars. The mean relative distance error between the IGS generated input burn scar source polygons and WSGA source polygons was 0.33 with a standard deviation of 0.44, while the mean relative simulation duration error was 0.18 with a standard deviation of 0.16. The variation was caused by some WSGA seeded wildfires underestimating or overestimating the number of source polygons that were used to create the input burn scars. Many of the WSGA seeded wildfires had relative simulation duration errors that were below the mean, which was inflated due to outliers such as Wildfire 6.

It was also clear that an increased fitness score does not always mean a lower relative simulation duration error and relative distance error, when comparing the WSGA to the IGS generated input burn scars. This can be seen in Wildfires 2 and 9, where Wildfire 2 has a fitness score of 0.97 while Wildfire 9 has a fitness score of 0.95. Wildfire 9 has a lower relative distance error and relative simulation duration error than Wildfire 2.

Only a single fire, Wildfire 12, had a perfect fitness score of 1 and the WSGA ended the genetic algorithm early, at generation 531. This occurrence could become more common by increasing the maximum number of generations. Even though Wildfire 12 had a perfect fitness score it still had a minor relative simulation duration error of 0.0008. This small relative simulation duration error still had a perfect fitness score because the IGS runs on a grid, therefore small changes in time may not affect the class of polygons. Both the IGS and WSGA simulations for Wildfire 12 had the same polygons in the same classes of ignited or vegetation at the end of their respective simulation duration when compared in the fitness function. From the perspective of the IGS, the WSGA seeded wildfire was essentially identical to the wildfire that created the input burn scar as the relative distance error between source polygons was 0. There is, however, a chance that two different wildfires (different source polygon locations and simulation durations) could produce the exact same wildfire burn scar, but this did not happen in the twenty generated wildfires.

The wildfire with the highest relative distance error between source polygons was Wildfire 6 with a relative distance error of 1.35. The WSGA seeded wildfire burn scar had a similar appearance to the input burn scar, but the WSGA overestimated the number of source polygons used to create the input burn scar (Figure 4.13). This is because the relative distance error is sensitive to errors in the number of source polygons, use of a different metric could be an alternative solution. Wildfire 5 had the highest relative simulation duration error of 0.49. This was due to the WSGA overestimating the number of source polygons. The results seem to indicate that wildfires with smaller areas tend to have a higher chance of the WSGA estimating more source polygons than necessary. This is most likely caused by how the fitness score is calculated, as smaller wildfires only cover a small region, meaning any small wildfire that the WSGA estimates close to the input burn scar will result in a high fitness score. This could be improved by changing the ignition and vegetation polygon weights for the fitness function, increasing the resolution of polygons near the burn scar and increasing the maximum number of generations. An additional step using a clustering algorithm such as K-means clustering could also be used to reduce this effect [166]. It could group source polygons by proximity, where only the polygon covering the centroid of that group is taken and used as a source polygon, the polygons that made up this group would no longer be source polygons. However, methods such as K-means

clustering could prevent the WSGA from estimating multiple source polygons that are near each other.

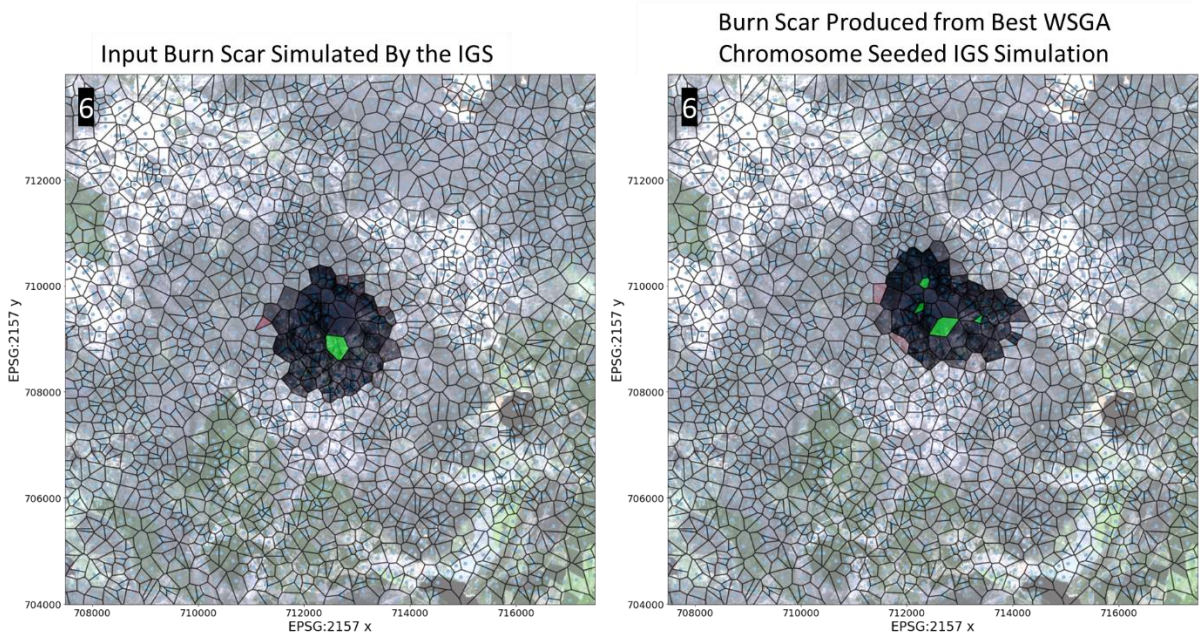

Figure 4.13: The input IGS burn scar for wildfire 6 (left) and the burn scar produced by the best chromosome created using the WSGA (right). The condition of the polygons is represented by their colour, not burnt (transparent), wildfire source (green), burning (red) and burnt (black) based on the mass of the polygon in that state, with a satellite image of the terrain in the background (Copernicus 2022).

From the results, properties such as the elevation map, land cover map, land cover fuel data and wind data do not seem to affect the WSGA's ability to accurately estimate the location of wildfire ignition points and propagation durations, but they do affect the IGS simulations. This is understandable as the terrain properties, weather conditions and IGS used to generate the input burn scars, were all also used to simulate the WSGA seeded chromosomes in the fitness function.

The WSGA may potentially be able to estimate the ignition points and propagation durations of real burn scars with higher precision given the availability of accurate weather data and the ability to model temporally variable weather data. Erroneous wind and terrain data have potential to negatively affect the WSGA's ability to accurately estimate the location of ignition points and propagation durations of real wildfires as the IGS uses the wind and terrain data during the WSGA's fitness function. Due to a lack of extensive open-source burn scar databases with accompanying ignition points in the region of interest, it

was not possible to compare the WSGA's estimated ignition points with real ignition points. However, it was possible to simulate WSGA chromosomes to produce burn scars using the IGS which could be compared to real wildfire burn scars. This lack of open-source data prevented the measurement of how errors in the weather data, terrain data or the wildfire simulator effect the WSGA. If such data became available and the WSGA had extensive testing, there would be an opportunity for it to act as a guidance tool for fire investigations or ecological research to identify ecosystems that are more prone to wildfire ignitions. For fire investigators, when analysing multiple real burn scars, some locations may have a repeated history of ignition points. It may be advantageous to consider removing forest litter or creating fire breaks in these areas to help prevent potential future wildfires.

The WSGA does not produce perfect results, and it may estimate ignition points and propagation durations which do not create a burn scar that exactly matches the input burn scar. Future work could include implementing a method to measure the uncertainty of WSGA source polygon and simulation duration estimates. Running the WSGA many times on the same IGS generated input burn scar can produce different relative distance errors and simulation duration errors. These varying values could be used to calculate the standard error of both relative distance errors and simulation duration errors which is one way of measuring uncertainty. An alternative approach could be to run the WSGA multiple times on the same IGS generated or real input burn scar and then produce a heat map of where the WSGA predicts the source polygons to be [153]. Most of these techniques involve running the WSGA many times which may be difficult due to the time required to compute results. It should be remembered that the program produces a list of elite chromosomes where only the chromosome with the highest fitness score is used. Lower ranking outputs may have merit and, in some applications and could be presented alongside the best estimate. Variability in the chromosomes within the elite list may indicate that many chromosomes can produce a satisfactory match to the burn scar. A fire investigator could then select which chromosome was the most likely to cause the burn scar.

4.5 Conclusions

This chapter presented an effective method for estimating the ignition points and propagation durations of wildfires, using a genetic algorithm called the WSGA. This method was able to perform the task of estimating the location of ignition points and propagation durations of wildfire, simultaneously using only the environmental data inputted into the IGS, and the wildfire burn scar as input. No additional data such as the number of ignition points were required.

It is important to note that the WSGA is an optimisation algorithm. Its accuracy is limited by the accuracy of the wildfire model and computation power. The WSGA is also limited by the number of polygons used for the IGS and the user defined maximum simulation duration. As the IGS runs on a finite contiguous grid of polygons the resolution of its output can only be as accurate as the size of the smallest polygon. The size of these polygons is determined by the density of polygons placed in the same local area. This made it difficult to estimate the correct ignition points and propagation durations of both an extremely small and extremely large wildfire using the same user defined parameters such as number of polygons in the grid. If a small number of polygons are used on a small burn scar, then the available polygons may not provide a high enough resolution to estimate a meaningful ignition point. If many polygons are used, then the WSGA can take an extremely long time to produce meaningful results. An increase in the number of polygons in the IGS simulation could also require an increase in the number of chromosomes in a generation for the WSGA to work effectively, further increasing the time required to compute the results. This could potentially be fixed by allowing cells to dynamically scale based on proximity to the inputted wildfire burn scar centroid. The user defined maximum simulation duration also sets an upper limit for how long an IGS simulated wildfire can propagate for. If the maximum simulation duration is lower than the propagation duration of the input burn scar, then the WSGA can never estimate the correct simulation duration. However, if the maximum simulation duration is too high then some chromosomes may have extremely long propagation durations which may slow down the program. It should be noted that if a wildfire completely burns a region surrounded by non-combustible material, any point in that region could have produced the resulting in the burn scar. The WSGA would not be

able to estimate the ignition points or propagation durations of such a wildfire. An active wildfire front will always provide a boundary that can be used to suggest ignition points. Human interventions may impact the WSGA's ability to accurately estimate the ignition points and propagation durations of wildfires if the interventions are not recorded in the forward running wildfire simulator and correctly reflected while simulating chromosomes during the fitness function.

The WSGA could be improved by adding or removing some of the steps in the genetic algorithm, detailed in Section 4.2. During the development of the WSGA, mutation involving generic bit-flipping was a step in the process after two-point crossover. It seemed to have a negative effect on the fitness scores of chromosomes while also not providing much diversity in the output. Therefore, it was replaced with the diversification step. Minor changes to the values of variables used in the genetic algorithm could also have potential to improve it.

It would be possible to modify the WSGA to estimate unknown environmental conditions. These could include the elevation map, land cover map, land cover fuel data and wind data. The values of these parameters could also vary temporally throughout a simulation. However, estimating these parameters increases the search space as the number of unknown variables greatly increases the number of possible states the wildfire simulation can be in. Therefore, the computing time would increase in a non-linear fashion.

Future work includes simulation studies where the optimal number of generations is assessed to balance a trade-off between accuracy and computational time. Additional tests could also be performed where the wind speed and direction vary throughout the IGS simulations. However, temporal changes in the weather would not affect the algorithm's ability to estimate the ignition point and propagation duration of a wildfire if the time that wind speed or direction changes is known and implemented in the forward running wildfire simulator. A different wildfire model could also be trialled which captures other wildfire behaviours such as spotting [17]. A new process to calculate the relative distance error between source polygons from the input wildfire burn scar and the WSGA seeded wildfire could be developed. Currently, the mean diameter of a wildfire scar used to calculate the

relative distance error between burn scars. This may not represent every wildfire perfectly due to the variability of wildfire shapes, so another metric could be devised. Future work could also include using wildfire risk maps to inform the WSGA of areas that are more likely to include potential ignition points, decreasing the search space and time to compute potential solutions [167].

If a dataset of real burn scars in the Wicklow Mountains, Ireland containing the ignition points and propagation durations of these burn scars were released the next step would be to examine the performance of the WSGA and the IGS using this real data. Controlled burning may also provide a source of this data. Other datasets in areas outside of Europe may provide a means of further verification for the WSGA [168]. Further studies using these datasets could also yield a method for measuring uncertainty when examining real wildfires for results produced by the WSGA.

The WSGA has demonstrated through the results to be an effective tool for estimating the ignition points and propagation durations of a wildfire. This study presents the first automated method with the goal of estimating the ignition points and propagation duration of a wildfire using wildfire burn scars. If the WSGA is presented with the ignition points, number of ignition points or the propagation duration as an input, it becomes a lot easier to computer the other values, but the WSGA can estimate all simultaneously. The method used in the WSGA is also not just constrained to estimating the ignition points of wildfires, but it also has potential to be used to estimate the sources of other propagating objects on a grid such as disease which could be explored in the future [144].

Chapter 5 Wildfire Modelling using Computer Vision and a Desktop Apparatus

Mathematical modelling is used as a common approach to forecast how a wildfire will spread. Wildfire models such as the Rothermel model were constructed using experimental data obtained by burning fuel arrays with an area between $2m^2$ and $4m^2$. This chapter describes the development of an experimental system to explore the burning behaviour of smaller fuel arrays consisting of sheets of paper treated with potassium nitrate. In total 123 burn sequences were performed to gather data [169]. An apparatus was developed to allow effective filming of paper burning in conditions where wind speed and orientation of the paper are controlled. When ignited, the treated paper creates a smouldering fire. The propagation rate of the fire is extracted from the RGB video using computer vision. While a computer vision pipeline was developed to extract data from the experimental results, the information gained from analysing these results is this chapter's primary novel contribution. Experiments were performed where the treated paper was burned at different angles (to investigate the effect of slope) and different wind speeds in a desktop wind tunnel. The propagation rates measured in these conditions were then analysed. The experiment performed in this chapter is best suited for educational and demonstrational purposes, but the experimental results were compared to a wildfire simulator running a fitted version of the Rothermel model.

Section 5.1 provides an overview of the apparatus used to perform these experiments. Section 5.2 discusses how the filter paper was treated with potassium nitrate. Section 5.3 describes the computer vision pipeline developed to track the fire line. Section 5.4 examines the effect of slope and wind speed on rate of spread. Section 5.5 compares the experimental data produced with a wildfire modelling framework, the Irregular Grid Software (IGS) (Chapter 3) running an altered version of the Rothermel model [157] [47]. Section 5.6 provides a discussion of the experimental results in this study. Section 5.7 states overall conclusions.

5.1 Apparatus

An apparatus was designed consisting of four main components: a rotatable mount for the filter paper, an igniter, an object for calibration with physical reference points and a wind tunnel (Figure 5.1).

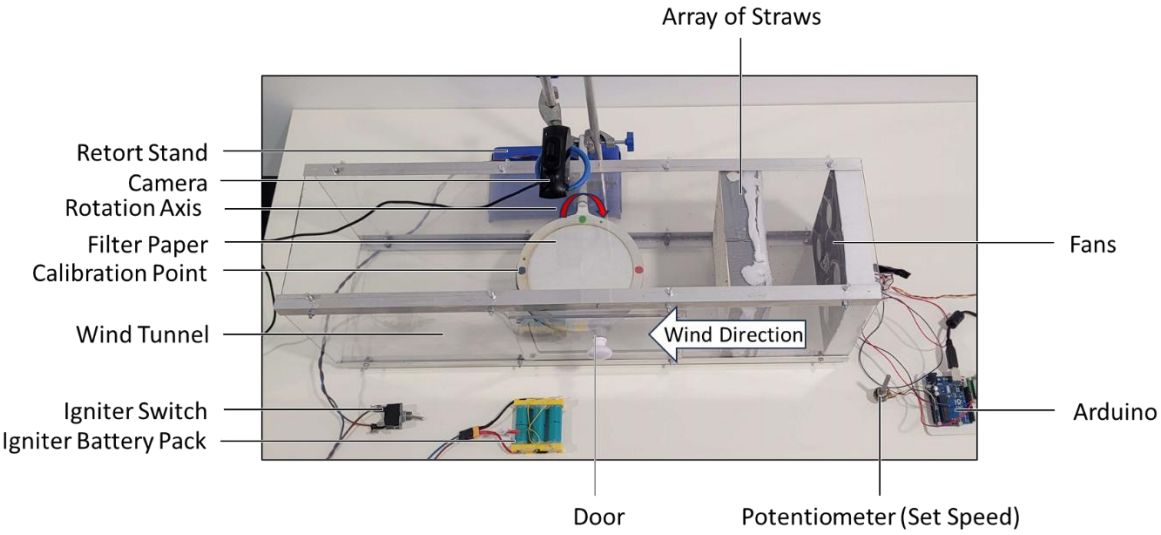

Figure 5.1: Rotatable filter paper mount inside the wind tunnel.

Filter papers were placed on top of the rotatable mount. The rotatable mount consisted of an aluminium ring (bottom ring) attached to an aluminium bar. The bottom ring was held in place by a bar using a combination of a retort stand and a bosshead. The bosshead and retort stand allowed the mount to be rotated around the roll axis. This allowed the filter paper to be titled at different angles during experiments which emulated sloped terrain. A Logitech HD C920 Pro Webcam (Carl Zeiss Tessar lens, HD 1080p) camera was also connected to the bar on an elevated platform facing the mount. When the mount was manually rotated before experimental runs the camera would also rotate by the same amount. This provided an overhead view of the filter paper regardless of slope angle. The camera recorded all filter paper burns that took place from ignition until the burnt area reached any section of the ring, or it extinguished. Filter paper was placed on top of the bottom ring. A disk of sheet metal with a small hole in the middle was attached below the bottom ring to hold the igniter in place. The disk did not touch the filter paper, but the igniter did through the small hole in the disk. The top of the bottom ring had two

orientation pins so an additional painted ring (top ring) could be aligned and placed on top to hold the filter paper (Figure 5.2). Clamps were also used to hold the top ring and filter paper in place for burns where the filter paper mount was held vertical at 90°.

The igniter was attached to the sheet metal under the bottom ring; the igniter protruded through the hole in the disk where it contacted filter paper placed on the bottom ring (Figure 5.2). The igniter was made using Kanthal® A-1 wire which was wound into a coil. This wire has high electrical resistance (approximately 1Ω), therefore once an electrical current is supplied, the wire will heat up rapidly causing ignition of the paper. A toggle switch was used to control current flow in the circuit (Figure 5.1). The circuit was powered using three lithium-ion cells wired in series, this produced 9.6 volts which had enough current to heat the wire (Figure 5.3). One charge of the battery was sufficient to start the burn over 100 times.

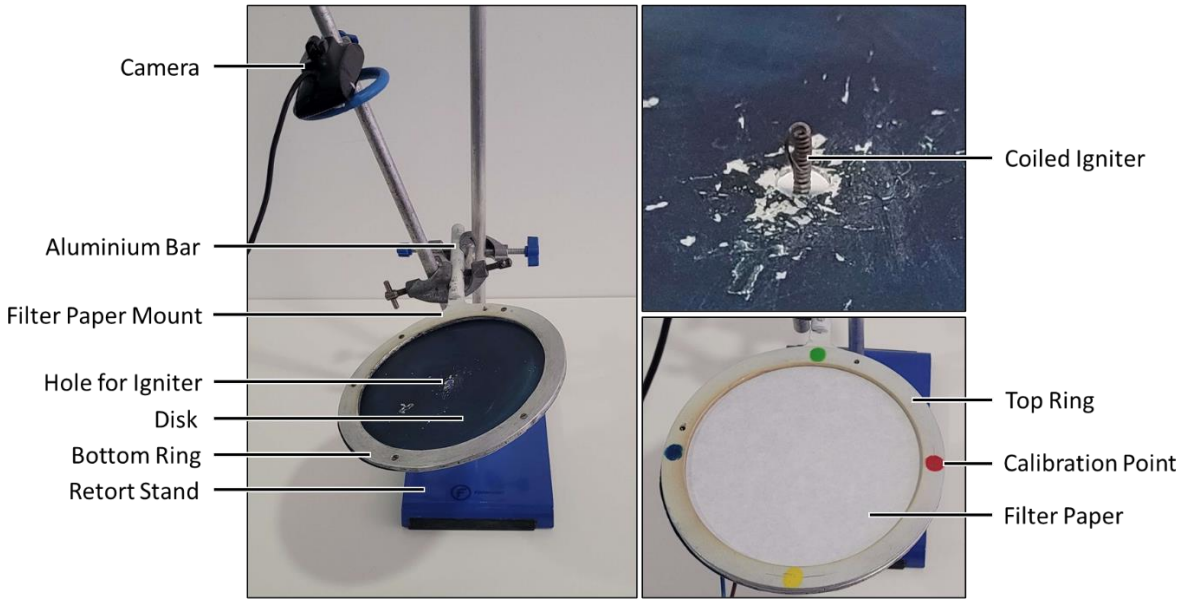

Figure 5.2: Left: Rotatable filter paper mount with camera attached. Top Right: Coiled Igniter protruding from disk. Bottom Right: Top ring with calibration points on filter paper.

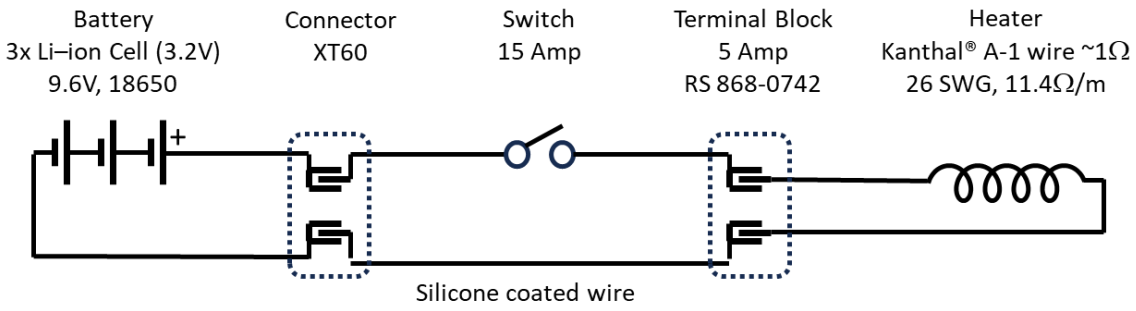

Figure 5.3: Circuit diagram of the igniter.

The painted top ring was coloured white using anticaloric (heat resistant) paint. Four differently coloured circles (calibration points) were then painted on the top ring using coloured anticaloric paint. The positions of these colours relative to the igniter for all experiments are as follows, blue was upslope with the wind, yellow was perpendicular (on the left) to the slope and wind, green was perpendicular (on the right) to the slope and wind, and red was down slope and against the wind (Figure 5.2). These physical reference points were used to calibrate SI units for measurement of position.

Four transparent acrylic sheets were used to construct a wind tunnel (with a length of 0.7m, width of 0.2m, and depth of 0.2m). Four CPU 4-wire (diameter of approximately 0.09m) fans were attached to one end of the wind tunnel to blow air through it. The fans ran at a fixed voltage (12 volts), and their speed was controlled using pulse width modulation (PMW). The duty cycle of the PMW was controlled using an Arduino Uno (Revision 1) with a potentiometer as input (Figure 5.4). An anemometer was used to measure wind speed. An array of straws was mounted in front of the fans as a flow straightener [170]. Existing experiments have shown that flow straighteners condition the air into laminar flow [171]. The straws were held together using varnish. There was a circular hole in the middle of one of the acrylic sheets that make up the wind tunnel; the bar from the filter paper mount slotted into this. This allowed placement of the top and bottom rings inside the tunnel where they could be rotated. There was also a small door on the acrylic sheet opposite the hole to allow manual adjustment of the filter paper mount (e.g. to apply and measure rotations) (Figure 5.1).

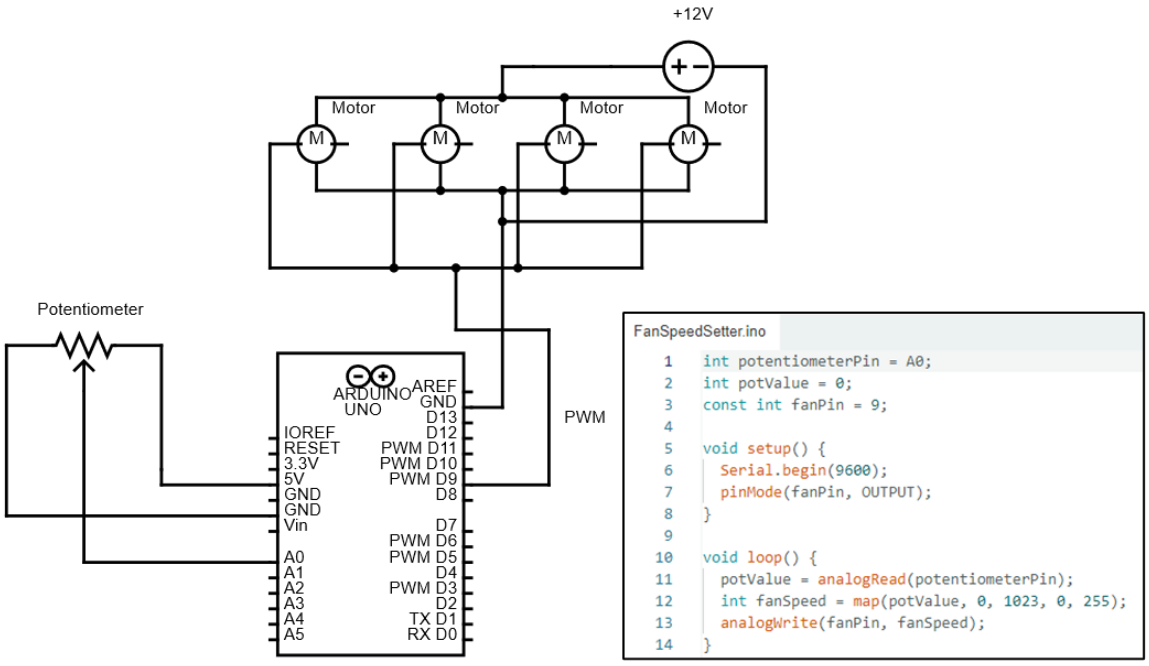

Figure 5.4: Circuit diagram of the four fans for the wind tunnel, where their wind speed is controlled by an Arduino Uno with accompanying code. A potentiometer is used as input.

All experiments were conducted in a fume cupboard with the sash closed. Experiments where wind was not applied were performed outside of the wind tunnel to prevent excess smoke blocking the camera's view of the filter paper. For these experiments the airflow in the fume cupboard was turned off. Experiments where wind was applied, were performed in the wind tunnel and airflow in the fume cupboard was turned on to reduce the amount of smoke diffusing into the room. For experiments outside the wind tunnel indoor lighting was used to light the scene. Indoor lighting was reflected on the acrylic surface of the wind tunnel and therefore natural daylight was used for experiments taking place within the wind tunnel. After each burn, the filter paper mount was cleaned.

Experiments involving wind speeds of 1m/s and greater were completed first and recorded using the Parkside Anemometer PWN A1. The Parkside Anemometer PWN A1 was unable to record wind speeds less than 1 m/s though, so the RS PRO IM-740 Rotary Vane Anemometer was used instead for these wind speeds. These anemometers were compared with each other in Section A.3 Anemometer Validation, Table A.1. Anemometer readings were made above top ring of the filter paper mount on the opposite side to the fans where the anemometer was held adjacent to the filter paper mount (on the side of the blue calibration

point). Angle of slope was recorded using a Parkside Digital Inclinometer PNM 2 A1, where recordings were made along the roll axis of the filter paper mount. The inclinometer was placed on the yellow calibration point from the top ring for all measurements.

5.2 Preparation of Filter Paper

Fisherbrand™ Grade 601 General Purpose Filter Papers were used in this experiment. They had a mean diameter of 0.15m [172], a mean thickness of 1.45×10^{-4}m (Standard Deviation (SD): 6.56×10^{-6}m) and a mean mass of 1.62×10^{-3}kg (SD: 2.51×10^{-5}kg). Potassium nitrate (KNO_3) was added to the filter paper for three reasons: to prevent a flaming fire, to control the burn rate and to ensure a continuous and uniform fuel bed [77] [76]. Preventing flames made performing computer vision at this scale easier, however the computer vision pipeline described in Section 5.3 was also tested on flaming fires of the same scale and functioned correctly most of the time. Therefore, it maybe be possible to expand the computer vision pipeline to work on real wildfires, but additional testing would be required. Once KNO_3 is heated it undergoes a decomposition reaction releasing potassium nitrite (KNO_2) and oxygen (O_2) at a mass ratio of 0.84:0.16 respectively (1kg of KNO_3 decomposes into 0.84kg of KNO_2 and 0.16kg of O_2):

$$2KNO_3 \xrightarrow[Heat]{} 2KNO_2 + O_2$$

(Eq. 5.1)

The heat released during combustion is sufficient for KNO_3 to decompose into O_2 which can accelerate the burn rate. It should be noted that the KNO_2 produced acted as an additional mineral, therefore slowing the burn rate.

To add KNO_3 to the filter paper, KNO_3 was dissolved into water at a concentration of 0.125kg/L (1.24 molar solution). The filter papers were then saturated in this solution and left to dry in the air. The mass of the filter paper was measured using an OHAUS© Pioneer™ PA512 scales with a readability of 1×10^{-5}kg [173]. Once this mass stopped decreasing it was considered dry and the moisture content was assumed to be the same as untreated filter paper which was 4.5%, this is discussed later in Section 5.5. Treated filter papers did not produce a flame but instead smouldered. Smouldering occurs when the process of

oxidation takes place on the fuel's surface, while flaming fires occur when the reactants for combustion are in a gaseous state [174]. It is possible that some smouldering fires could change to a flaming fire if the gases released during smouldering are ignited. Typically, smouldering fires spread at a rate approximately 10 times slower than flaming fires over a solid [175]. There are multiple reasons why KNO_3 may have caused smouldering to take place such as the additional mineral content from the KNO_2, that KNO_3 may be letting the process of combustion occur at a lower temperature therefore not igniting the gases produced from smouldering, or a layer of carbon dioxide (CO_2) could be preventing some of the O_2 in the air from interacting with the combustion process. Further studies would need to be completed to provide an exact reason. Treating the filter papers with KNO_3 gave them an additional mean mass of 4.11×10^{-4}kg (SD: 2.47×10^{-5}kg) for a total mean mass of 2.031×10^{-3}kg (SD: 4.12×10^{-5}kg).

5.3 Computer Vision Pipeline

A typical computer vision pipeline was used to automate analysis of the data [176]. The pipeline contained the classic computer vision steps of capture, enhance, segment and measure [113].

5.3.1 Capture

The camera recorded each filter paper burn. Recording started just before ignition at the centre of the filter paper, until the burned area reached the edge of the ring, or self-extinguished. Recorded videos were later trimmed to ensure the video began and ended within a few frames of these events. The frames all had dimensions of 1152x648 pixels. The first frame of each video was used to calibrate the sequence to find the parameters to transform, rotate and crop the following frames to measure burn distances (Figure 5.5). The remaining frames in the video sequence were then processed using these parameters. These steps are described below. All computer vision was performed using OpenCV in Python [109].

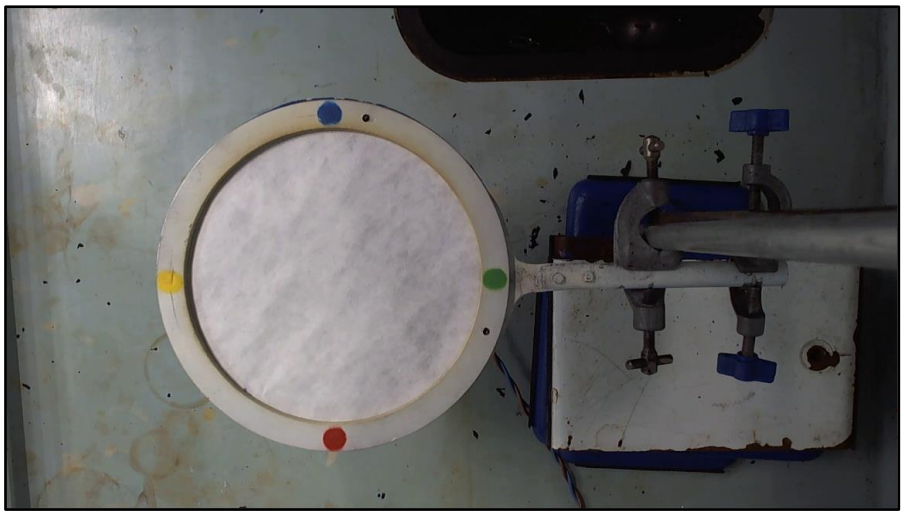

Figure 5.5: First frame of a selected video.

5.3.2 Enhance

This section describes how the first frame of a video sequence is used to calibrate the image processing pipeline and how they are applied to the following frames (Figure 5.6).

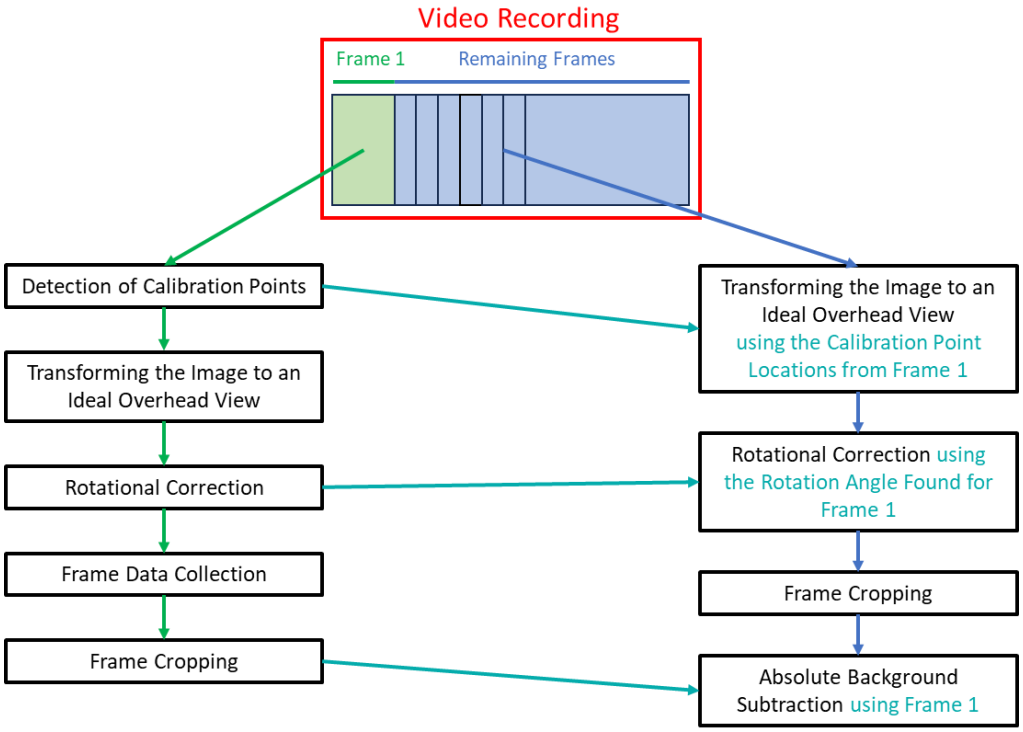

Figure 5.6: The steps used for frame enhancement. Processes that have green arrows pointing at them occur on the first frame. Processes with the blue arrows pointed at them occurred on every subsequent frame. The cyan arrows represent data from calibration processes on the first frame being used in subsequent frames.

Detection of calibration points was the first step of enhancement. The pixel positions of the coloured circles on the top ring were located. Coloured circles were used to simplify identification of each of the four points required for calibration. Full camera calibration using for example Zhang's method was not performed as calibration using the four points was sufficient to make the measurements required [177]. Different coloured circles allowed for each of the calibration points to be identified separately. To detect the coloured circles, the frame was converted from RGB format to HSV format for analysis. Different HSV ranges for each of the coloured circles were manually trialled to find the HSV ranges where detection of calibration points could be performed on all videos. Pixels in the HSV range of (90, 38, 30) to (174, 216, 128) were flagged as blue, (20, 55, 114) to (35, 225, 246) were flagged as yellow, (40, 38, 70) to (75, 216, 113) were flagged as green and both (0, 38, 76) to (10, 216, 148) along with (175, 38, 76) to (180, 216, 148) as red. The pixels identified were used to create masks for each respective colour (both sets of red flagged pixels were combined into one mask). A series of dilation and erosion kernels were performed on the mask to remove noise while retaining the position of the coloured circles. The sizes of these kernels were chosen by user input. The following kernel sizes detected all coloured circles and were therefore used. Dilation with a kernel size (3, 3), followed by erosion with a kernel size (20, 20) and dilation with a kernel size (19, 19). The masks were then duplicated and combined into one mask which was then inverted. Blob detection was performed on this combined mask. Different blob detection parameters were manually trialled to find the parameters where detection of calibration points could be performed on all videos. These parameters were: blobs of areas greater than 220 pixels, minimum convexity of 0.6, minimum circularity of 0.2 and minimum inertia ratio of 0.2.

The list of blobs was compared to the different coloured masks from earlier which provided a method of checking the colour of each of the four blobs. The correct pixel positions for each coloured circle were manually recorded for one frame. These manually recorded coordinates were used for all 123 videos, where the blob of the corresponding colour nearest these manually saved coordinates was saved (Figure 5.7). From this point the coordinates of the coloured circles will be referred to as calibration points.

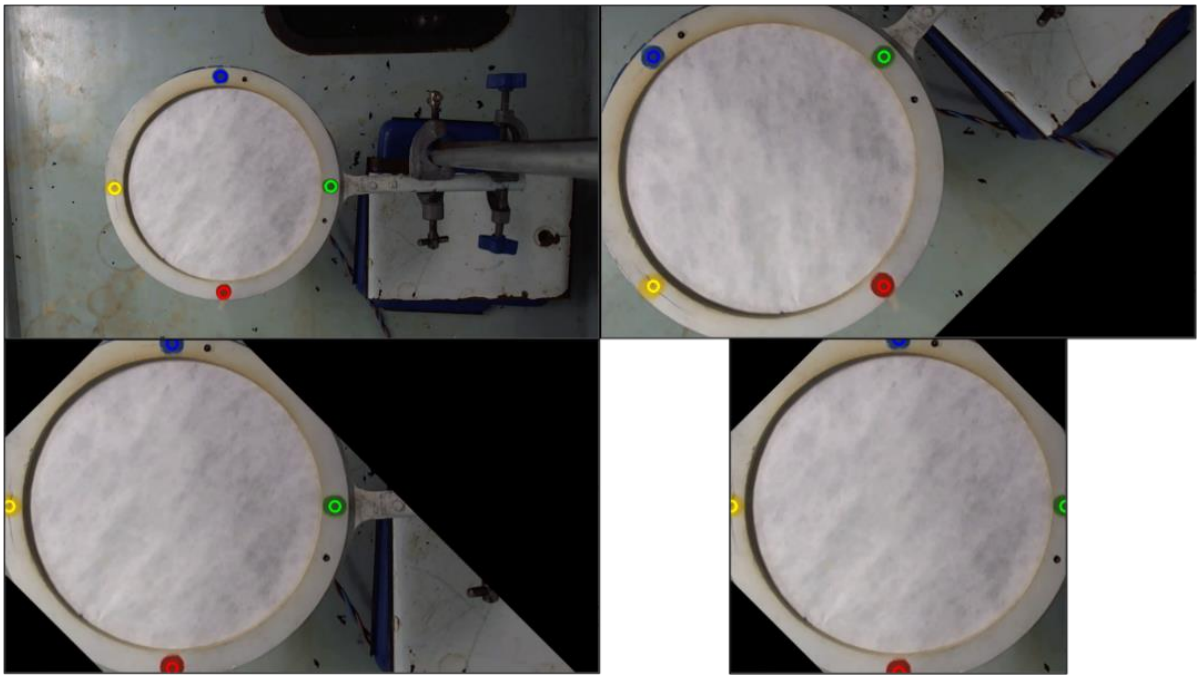

Figure 5.7: Processing applied to the first frame of a selected video during the enhancement stage of the computer vision pipeline. Top Left: Detection of calibration points on the top ring. Top Right: Overhead view transformation applied to frame. Bottom Left: Rotation transformation applied to the frame to position the blue, yellow, green, and red calibration points at the top, left, right and bottom sections of the frame respectively. Bottom Right: Cropped frame with the calibration points at the top, left, right, and bottom pixel coordinates of the frame.

Transforming the image to an ideal overhead view was the next step of enhancement. The calibration points detected in the previous steps were used to transform the image so that the filter paper and the top ring appeared as if viewed from directly overhead. The raw videos are almost, but not exactly overhead. This was done using a 2D homography, mapping the image to an idealised overhead plane [178] [179]. The homography mapped the calibration points to a new frame of reference with pixel coordinates of (100, 100), (100, frame height – 100), (frame width – 100, 100) and (frame width – 100, frame height – 100) for the blue, yellow, green and red painted calibration points respectively (Figure 5.7).

Rotational correction was the following step of enhancement. The frame was rotated so that the blue, yellow, green, and red painted calibration points appeared on the top, left, right, and bottom positions on the frame respectively. The angle of rotation required to do this was found, by calculating the $\tan^{-1}$ of the slope from the yellow calibration point to the green calibration point (no slopes in the 123 videos were undefined). A rotation matrix was used to rotate the calibration points to a position aligned with the North, South, East and West axes of the apparatus (Eq. 5.2) (Figure 5.7):

$$R_M = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix},$$

(Eq. 5.2)

where R_M is the rotation matrix and θ is the angle of rotation required to have the blue, yellow, green and red painted points in the correct positions. A homography was performed using the calibration points to rotate the entire image. The rotation aligned the image with the experimental geometry based on direction of wind and slope.

Frame data collection was then performed as part of enhancement. Data from the first frame were collected and used for analysis of the following frames. The igniter for the filter paper was positioned in the centre of the top ring. The ignition point was estimated as the intersection of the lines drawn between the blue and red calibration points, and green and yellow calibration points. This value was saved for the segmentation process. The distance between the centres of the yellow and green (same as distance between red and blue) painted circles on the top ring were measured at 0.1533m. This distance was also measured in pixels within the frame. The measurements were used to convert distances from pixel measurements to real distance in SI units (m).

Frame cropping was the next step of enhancement. The frame was cropped where the blue, yellow, green, and red calibration points represented the up, left, right and bottom coordinates in the frame respectively (Figure 5.7). This step increased processing performance and removed noise from the scene. The cropped frame was saved and used for comparison with the remaining frames in the video to detect burning. This was done using OpenCV's absolute difference function to perform background subtraction [180].

Processing of remaining frames in the video was the final step in enhancement. Each of the subsequent frames in the video were selected in turn. The same calibration parameters found for the first frame were used to transform, rotate and crop each subsequent frame. Background subtraction was used to identify regions of change between the current and first frame (Figure 5.8). This change was typically caused by the filter paper burning. Changes caused by smoke were ignored by converting the current frame to a HSV image

where pixels in the HSV range of (11, 45, 63) to (22, 128, 133) were found and excluded as they were likely to be smoke.

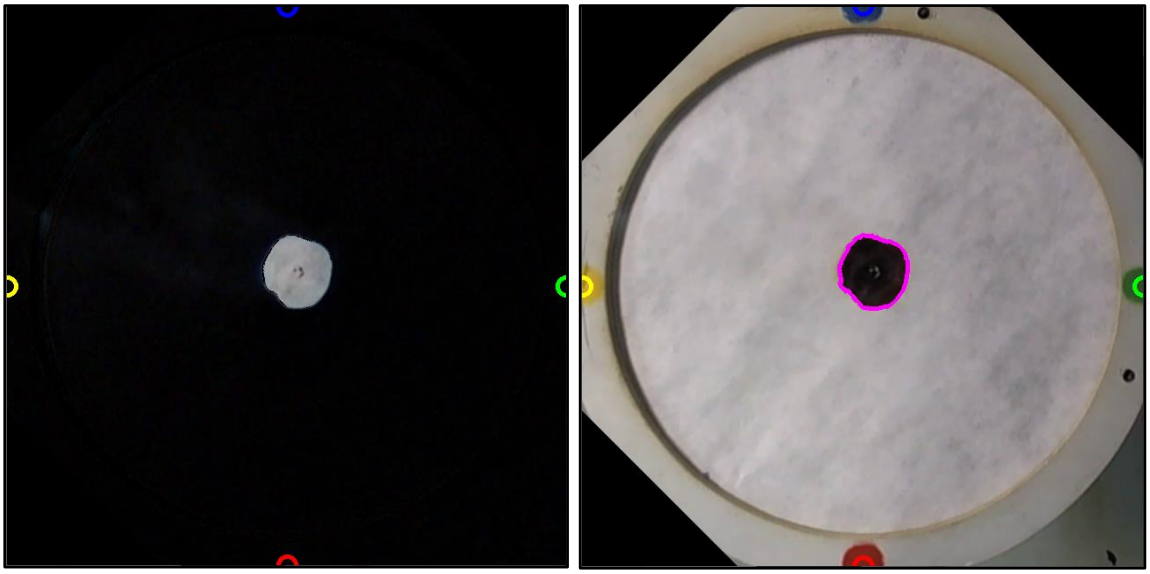

Figure 5.8: Left: Example of background subtraction performed between frame 400 and frame 1 of a video. The pixels coloured black represents areas where there has been minimal change. The pixels on the left image that look like those from the initial frame (coloured white in this case) represent areas where there has been a lot of change between the frames. Right: Contour (magenta line) drawn on the perimeter of the burned area on frame 400 of a video.

5.3.3 Segmentation

Contour detection was used to locate the perimeter of the burned area in each frame of video. There are different approaches to contour detection as seen in [181] and [182]. However, the algorithm presented in [183] was chosen as it is used in OpenCV's findContours() function by default [184]. Each frame obtained using background subtraction with the first frame (no burning) was converted to a greyscale image. Binary thresholding was applied for the greyscale values of 50 to 255. Contour detection was then used to create a list of contours using simple chain approximation where only the external contours were returned [185].

All contours were then parsed, and their centroid was compared with the centroid of the fire line's contour from the previous frame (the approximate ignition point found in the first frame was used for the following frame in the segmentation process). This process of updating the contour only occurs during any of the three following conditions, a contour has

not yet been detected in any frame from the video, the current frame number is less than 90, or the current contour has an area greater than 0.9 times that of the previous contour's area. Otherwise, the contour stays the same as it did in the last frame a valid contour was selected (if one exists). The first two conditions allow for larger variability in detected contours at the start of a video. The last condition expects the contour to generally grow in area over time (with a small margin of error). The contour that satisfies these conditions with a centroid nearest the location of the centroid from the previous frame's fire line contour was selected. This produces a time series of fire line contour growth but not for every frame. The selected contour was drawn on the frame from the video and used for visual confirmation of the process (Figure 5.8).

Straightforward contour detection using background subtraction worked as fire and smoke were the only moving objects in the frame.

5.3.4 Measurement of Burn Distances

Data on fire progression relative to the ignition point in the directions of up, left, right and down were recorded every 20 frames. These distances were found by getting the bounding box of the contour representing the burnt area. The difference in the vertical pixel coordinates between the ignition point with the top and bottom sides of the contour were found along with the difference in horizontal pixel coordinates between the ignition point with the left and right sides of the contour. These differences were then converted to SI units (m). It was also possible to record the burn distances in any other sets of four directions by rotating the frame to the desired angle before analysis.

5.3.5 Experimental Parameters

Burns were conducted with different wind speeds and slopes. Eight treated filter papers were burned with no wind and no slope. Five treated filter papers were burned with different combinations of wind speeds and slope angles. These burns included seven different non-zero wind speeds with no slope, seven different non-flat slopes with no wind

and nine different combinations of non-flat slopes and non-zero wind speeds (Table 5.1). This totalled 123 burns [169].

Wind Speed (m/s) / Angle of Slope (°D)	0	0.5	1	2	3	4	5	6
0	8	5	5	5	5	5	5	5
10	5	-	-	-	-	-	-	-
20	5	-	-	5	-	5	-	5
30	5	-	-	5	-	5	-	5
45	5	-	-	5	-	5	-	5
60	5	-	-	-	-	-	-	-
75	5	-	-	-	-	-	-	-
90	5	-	-	-	-	-	-	-

Table 5.1: Number of experimental burns performed for different wind speed and angles of slope.

In Section 5.4 the mean fire propagation rates with one standard error (SE) in different directions for different wind speeds and slopes were individually recorded. Linear regression was performed to get slopes and intercepts with one standard error for trend lines measuring how wind speed and angle of slope effect the propagation rate in the different directions. A t-value was also found so 95% confidence intervals can be calculated for the slope and intercept of these trend lines. The mean fire propagation rates with one standard error were also graphed in different directions for different wind speeds and slopes simultaneously.

In Section 5.5 the mean fire propagation rates with one standard error travelling upslope in the same direction as the wind for different wind speeds and slopes were individually compared to a fitted version of the Rothermel model. The mean fire propagation rate was also compared to the IGS, a wildfire modelling framework running the fitted Rothermel model (Chapter 3). This comparison was performed visually and mathematically using a threat score.

5.4 Results of Experiment

The distance that the fire had propagated over time for each of experimental burns at different wind speeds and angles of slope was recorded (Figure 5.9).

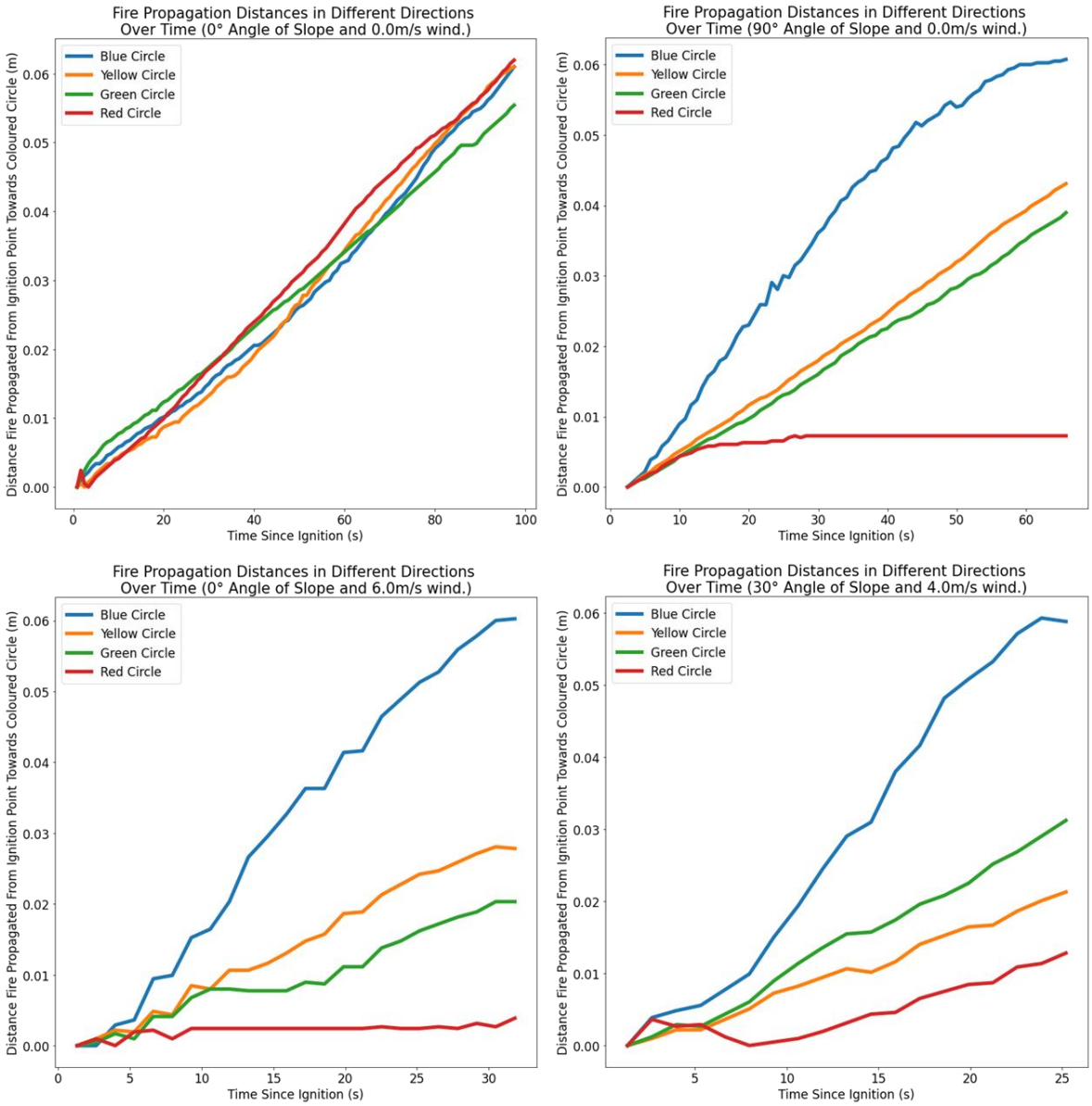

Figure 5.9: Distances of fire propagation over time recorded in different directions with four different combinations of wind and slope from selected video sequences.

The data provided the measurement and associated error for each wind speed and slope chosen. The mean propagation rate of the fire with one standard error in the direction of the blue (up slope, with wind), yellow (perpendicular to slope and wind on the left), green (perpendicular to slope and wind on the right) and red (down slope, against wind) calibration points for different wind speeds and angles of slope were evaluated.

The effect that different wind speeds had on the mean propagation rate of the fire in the selected four different directions with added trend lines are as follows (Figure 5.10).

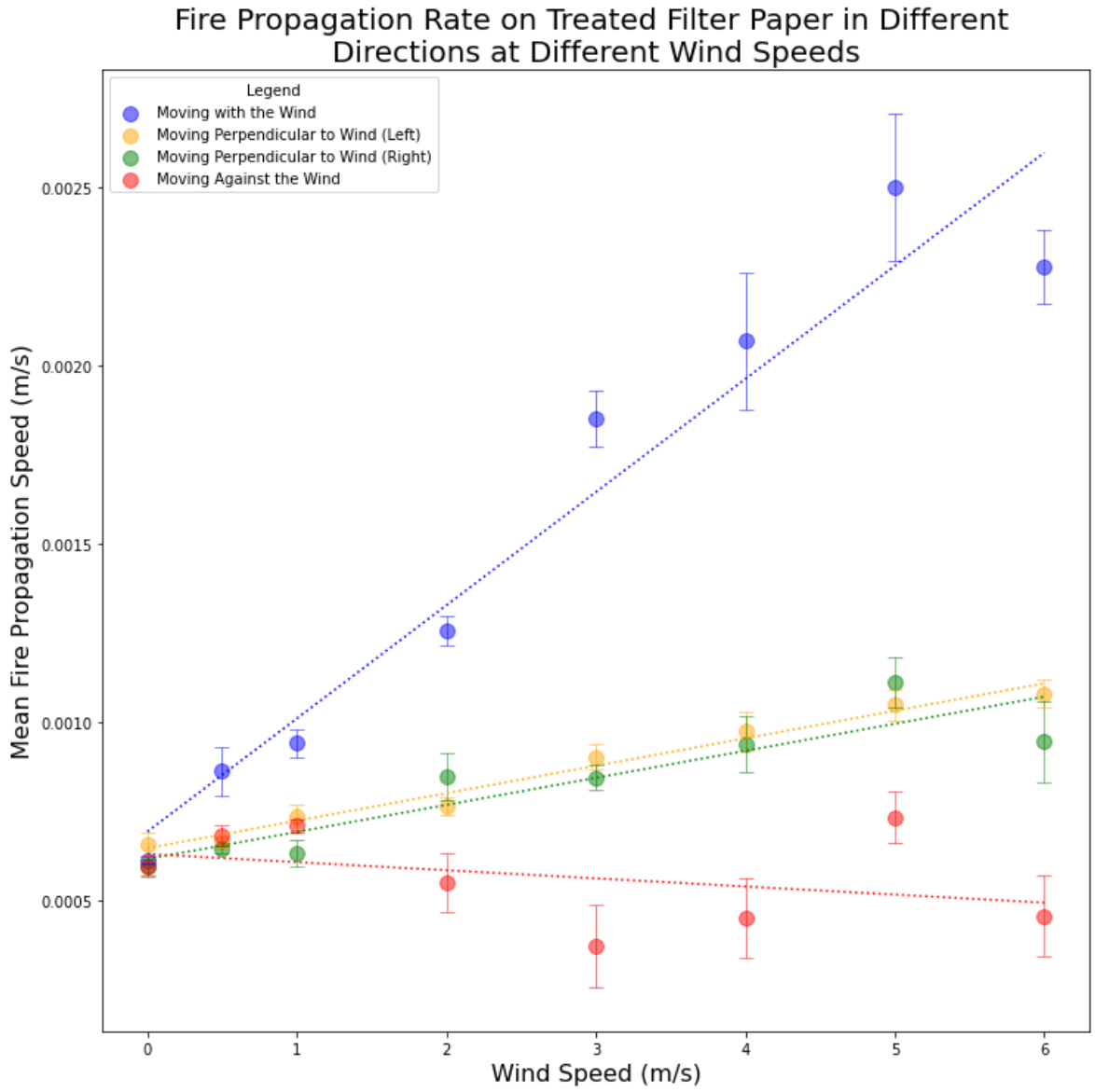

Figure 5.10: Mean fire propagation rate with bars representing one standard error for treated filter paper in the selected directions with no slope for different wind speeds with accompanying trend lines fitted using linear regression. For the listed cases the slopes and intercepts of the trend lines moving with the wind were 3.17×10^{-4} (SE: 3.3×10^{-5}), and 6.94×10^{-4} (SE: 1.12×10^{-4}), perpendicular to the wind (left) were 7.7×10^{-5} (SE: 4×10^{-6}), and 6.48×10^{-4} (SE: 1.5×10^{-5}), perpendicular to the wind (right) were 7.6×10^{-5} (SE: 1.4×10^{-5}), and 6.17×10^{-4} (SE: 4.8×10^{-5}), and against the wind were -2.3×10^{-5} (SE: 2.3×10^{-5}), and 6.31×10^{-4} (SE: 7.9×10^{-5}) respectively. A 95% confidence interval can be calculated by multiplying the SE of the slopes and intercept by the t-value of approximately 2.365. The eight sample points for fire propagating at each direction acted as seven degrees of freedom when calculating the t-value.

The effect these angles of slope had on the mean propagation rate of the fire in the selected four different directions with added trend lines are as follows (Figure 5.11).

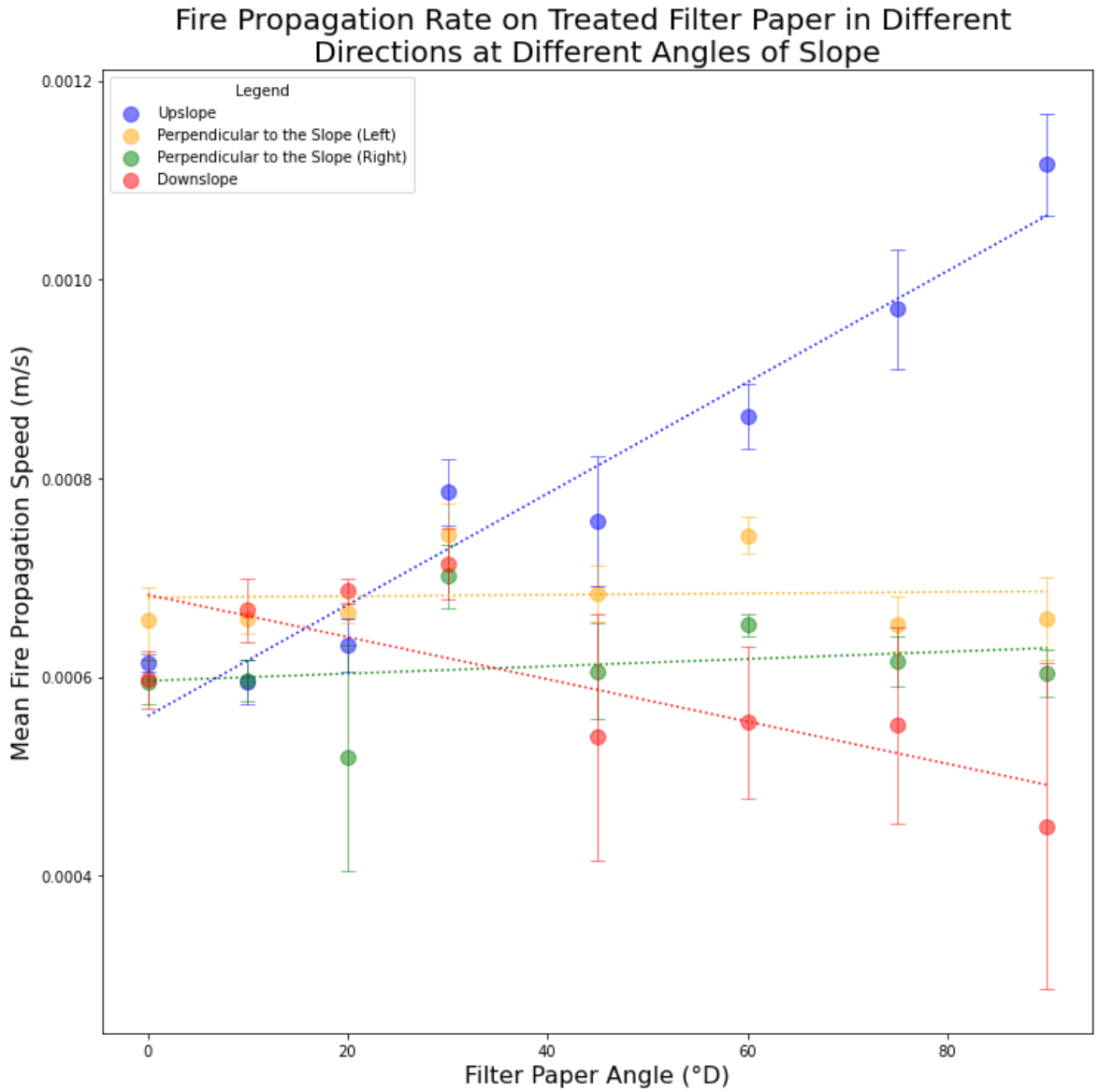

Figure 5.11: Mean fire propagation rate with bars representing one standard error for treated filter paper in the selected directions with no wind for different angles of slope with accompanying trend lines fitted using linear regression. For the listed cases the slopes and intercepts of the trend lines upslope were $6x10^{-6}$ (SE: $1x10^{-6}$), and $5.61x10^{-4}$ (SE: $3x10^{-5}$), perpendicular to the slope (left) were 0 (SE: 0), and $6.8x10^{-4}$ (SE: $2.5x10^{-5}$), perpendicular to the slope (right) were 0 (SE: $1x10^{-6}$), and $5.96x10^{-4}$ (SE: $3.3x10^{-5}$), and downslope were $-2x10^{-6}$ (SE: $1x10^{-6}$) and $6.83x10^{-4}$ (SE: $3.8x10^{-5}$) respectively. A 95% confidence interval can be calculated by multiplying the SE of the slopes and intercept by the t-value of approximately 2.365. The eight sample points for fire propagating at each direction acted as seven degrees of freedom when calculating the t-value.

The mean propagation rate for each combination of wind speed and angle from Table 5.1 was graphed in Figure 5.12, Figure 5.14, Figure 5.16, and Figure 5.18 with their standard errors in Figure 5.13, Figure 5.15, Figure 5.17, and Figure 5.19 respectively. Observed values are denoted by a black circle and cubic interpolation was used [186]. Mean propagation rate is denoted by darkness of colour and contour lines.

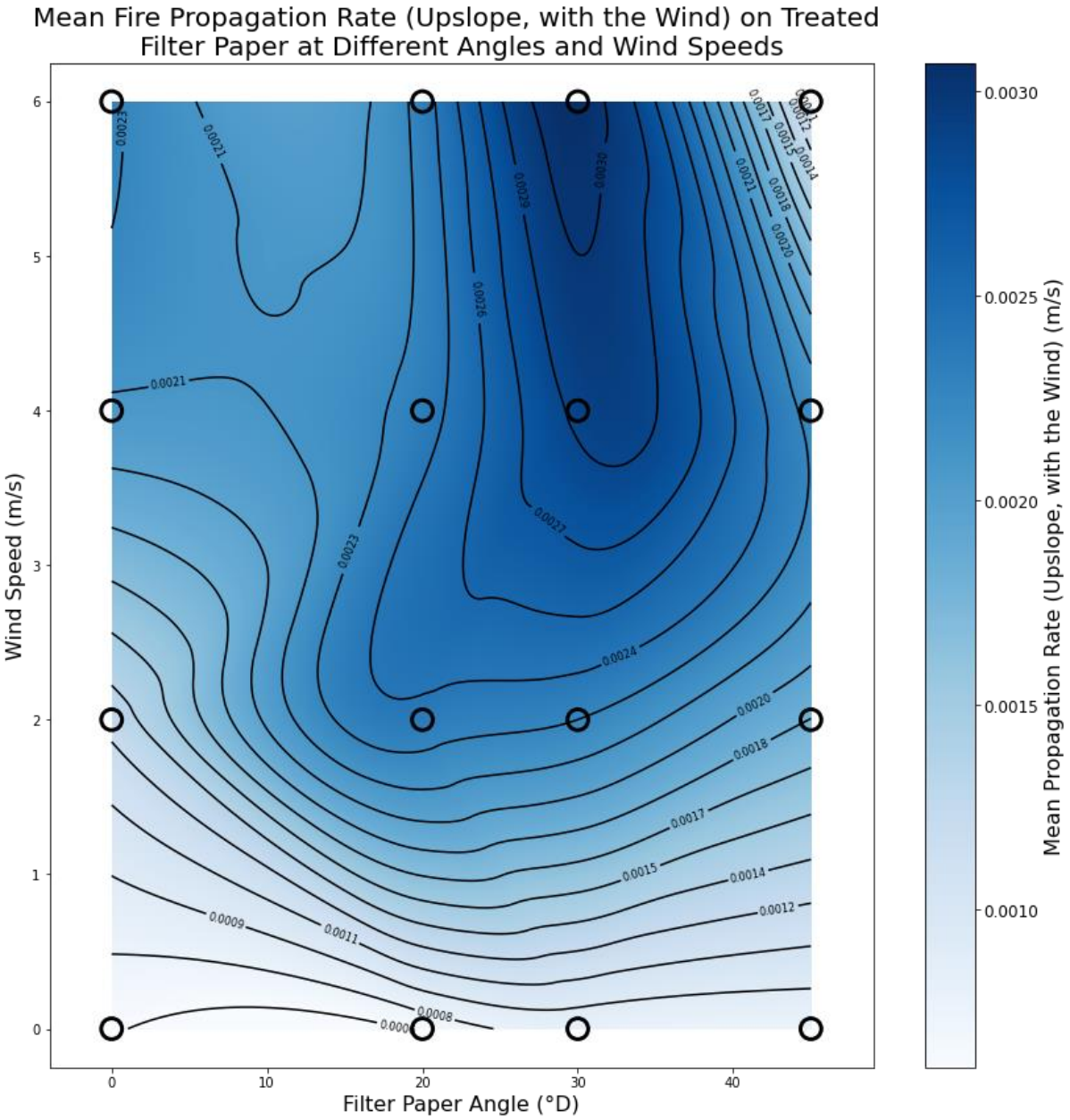

Figure 5.12: Mean propagation rate travelling upslope, with the wind on treated filter paper. Different wind speed and angle combinations were examined. Black circles represent observed data points and cubic interpolation was used. Each observed data point had five samples excluding the data point with no slope and wind which had eight samples. Mean propagation rate is denoted by darkness of colour and the contour lines. The fastest mean propagation rate occurs at an angle of 30° with a wind speed of 6m/s. The slowest mean propagation rate occurs at an angle of 0° with a wind speed of 0m/s. Some fires extinguished near angles of 45° and wind speeds of 6m/s.

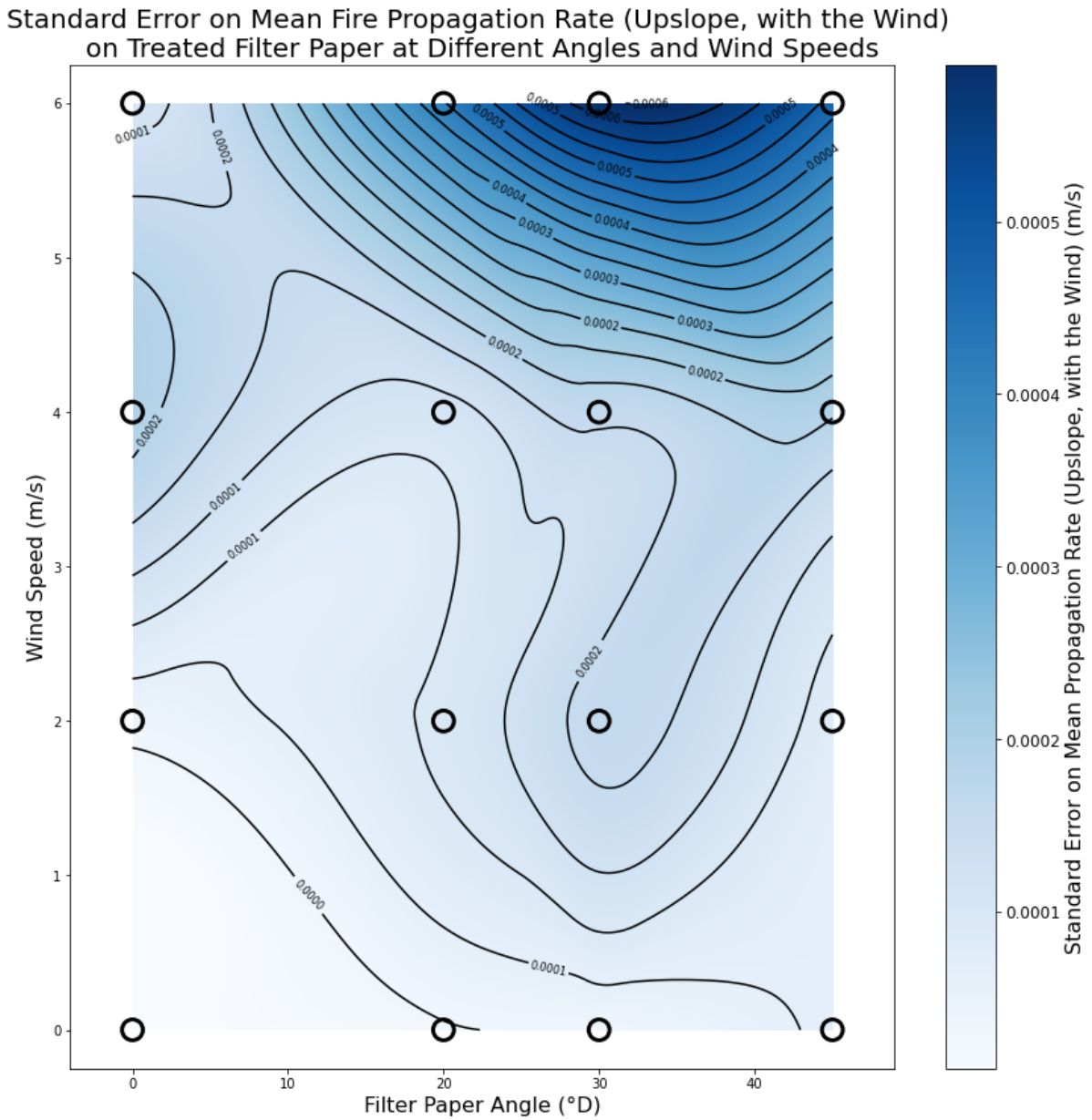

Figure 5.13: Standard Error on the mean propagation rate travelling upslope, with the wind on treated filter paper. Different wind speed and angle combinations were examined. Black circles represent observed data points and cubic interpolation was used. Each observed data point had five samples excluding the data point with no slope and wind which had eight samples. Standard Error on mean propagation rate is denoted by darkness of colour and the contour lines. The highest standard error on mean propagation rate occurs near an angle of 30° with a wind speed of 6m/s. The lowest standard error on mean propagation rate occurs at an angle of 0° with a wind speed of 0m/s. Some fires extinguished near angles of 45° and wind speeds of 6m/s which may be why this area has a higher standard error.

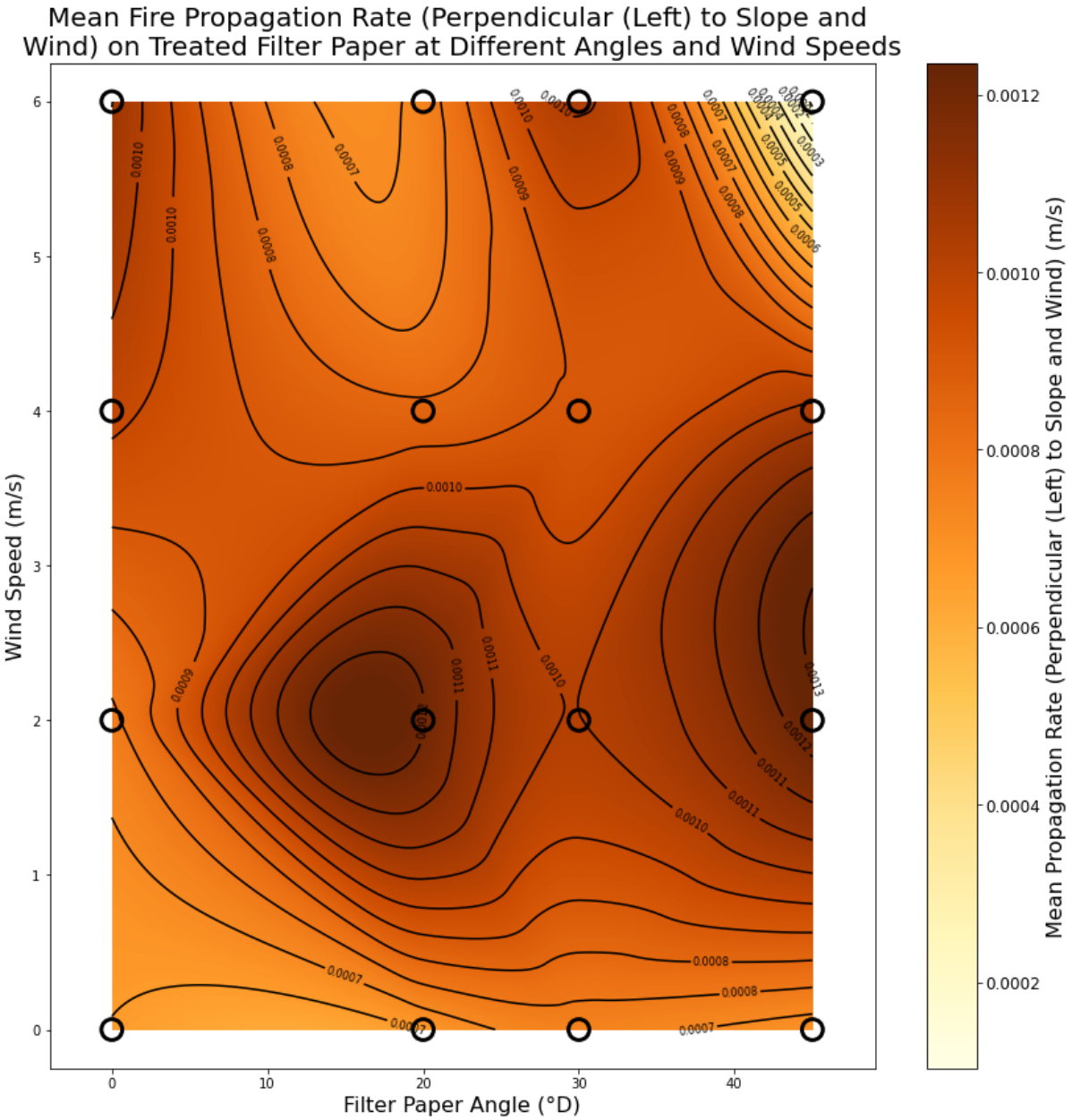

Figure 5.14: Mean propagation rate travelling perpendicular (left) to the slope and wind on treated filter paper. Different wind speed and angle combinations were examined. Black circles represent observed data points and cubic interpolation was used. Each observed data point had five samples excluding the data point with no slope and wind which had eight samples. Mean propagation rate is denoted by darkness of colour and the contour lines. The fastest mean propagation rate occurs at an angle of 45° with a wind speed of 2m/s. The slowest mean propagation rate occurs at an angle of 45° with a wind speed of 6m/s. Some fires extinguished near angles of 45° and wind speeds of 6m/s. The observed mean propagation rate at an angle of 20° with a wind speed of 2m/s is high, therefore creating a saddle point. This area has a high standard error on mean propagation rate as seen in Figure 5.15. This is not seen in Figure 5.16 and Figure 5.17.

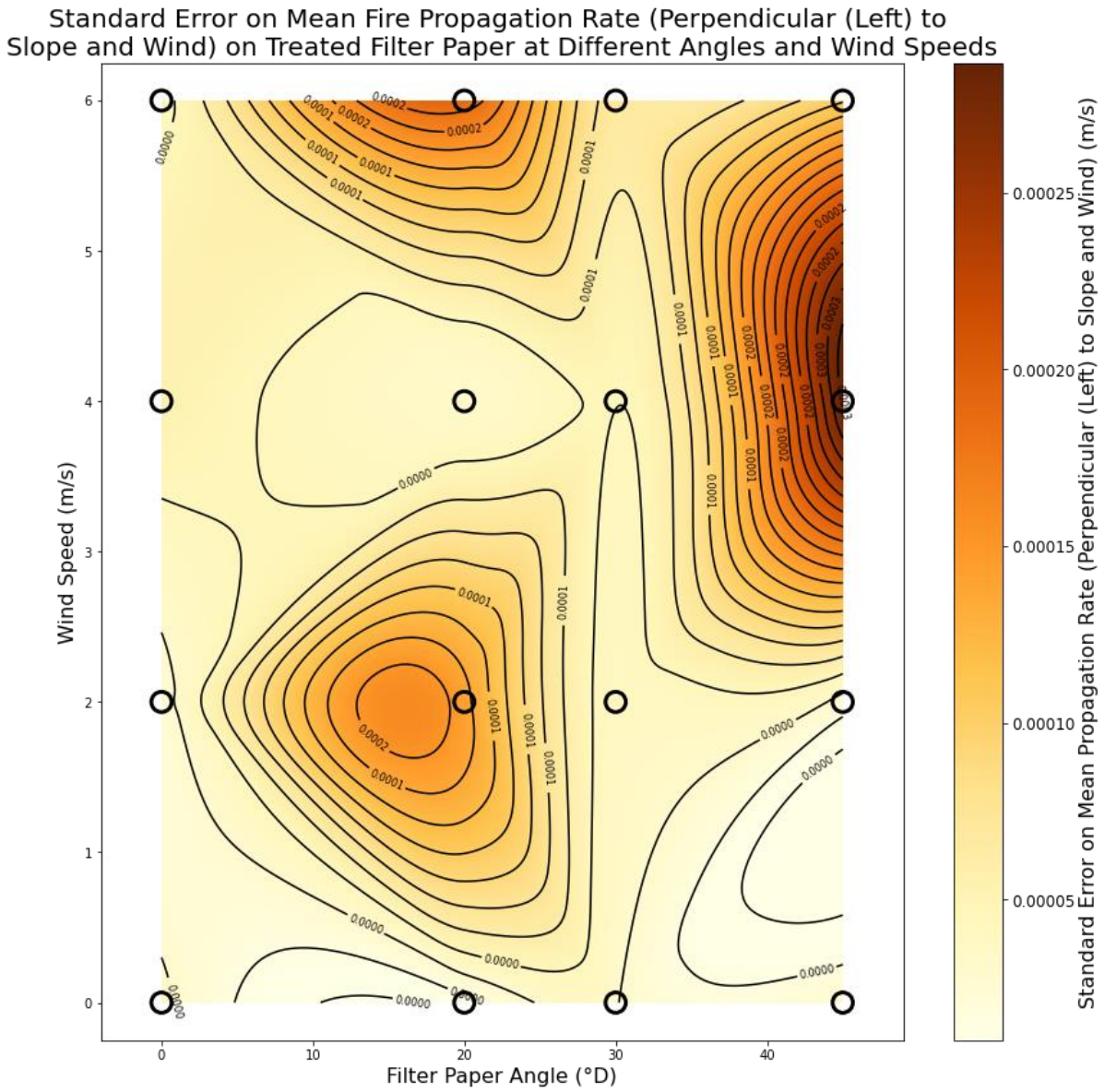

Figure 5.15: Standard Error on the mean propagation rate travelling perpendicular (left) to the slope and wind on treated filter paper. Different wind speed and angle combinations were examined. Black circles represent observed data points and cubic interpolation was used. Each observed data point had five samples excluding the data point with no slope and wind which had eight samples. Standard Error on mean propagation rate is denoted by darkness of colour and the contour lines. The highest standard on error mean propagation rate occurs near an angle of 45° with a wind speed of 4m/s. The lowest standard error on mean propagation rate occurs at an angle of 20° with a wind speed of 0m/s. A high standard error on mean propagation rate can be seen at 20° with a wind speed of 2m/s which is the same area as the saddle point from Figure 5.14.

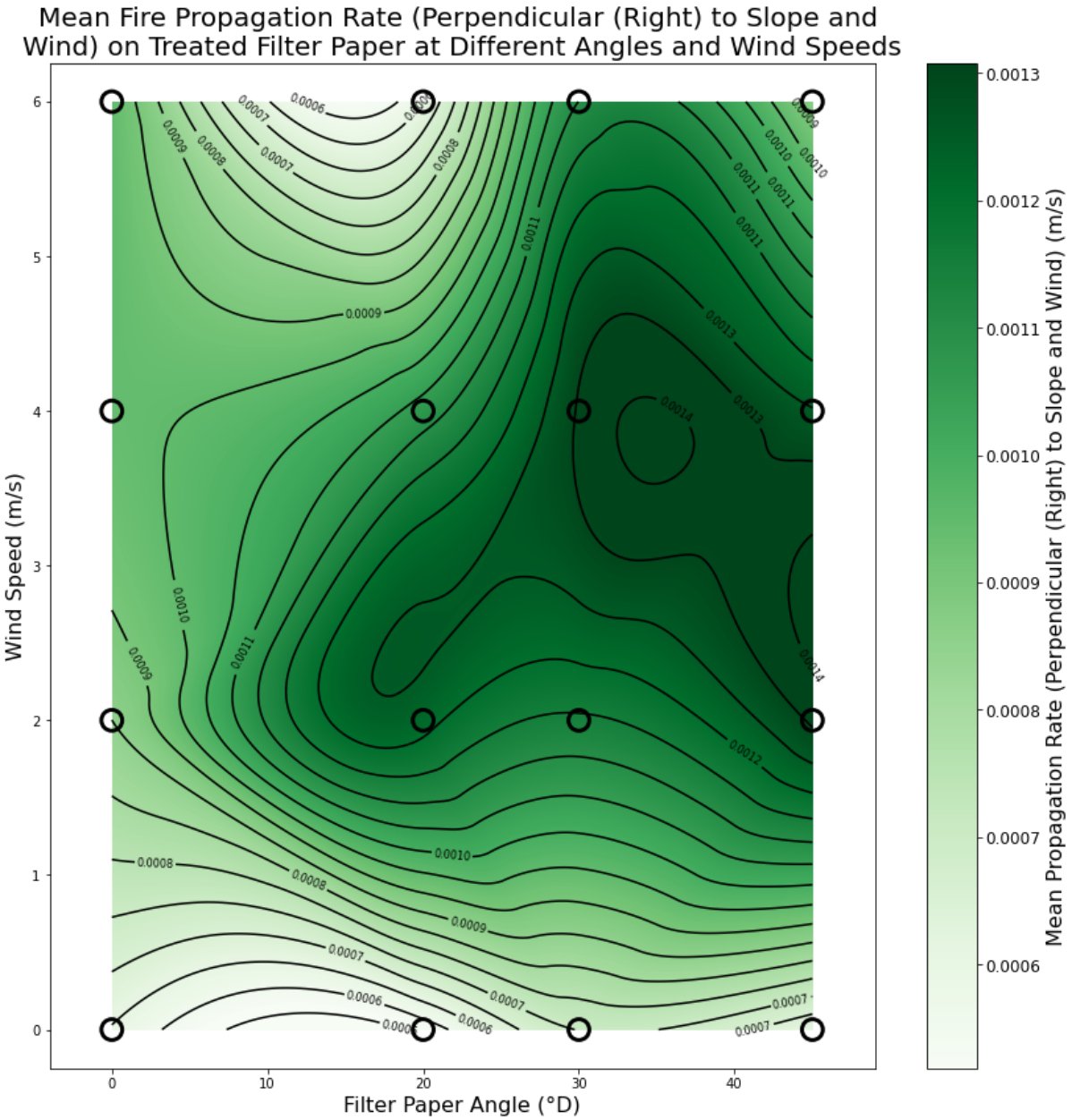

Figure 5.16: Mean propagation rate travelling perpendicular (right) to the slope and wind on treated filter paper. Different wind speed and angle combinations were examined. Black circles represent observed data points and cubic interpolation was used. Each observed data point had five samples excluding the data point with no slope and wind which had eight samples. Mean propagation rate is denoted by darkness of colour and the contour lines. The fastest mean propagation rate occurs at an angle of 45° with a wind speed of 2m/s. The slowest mean propagation rate occurs at an angle of 20° with a wind speed of 0m/s. Some fires extinguished near angles of 45° and wind speeds of 6m/s. Most of the recorded propagation rates are like Figure 5.14 which is to be expected as the propagation rate should be independent of the perpendicular direction.

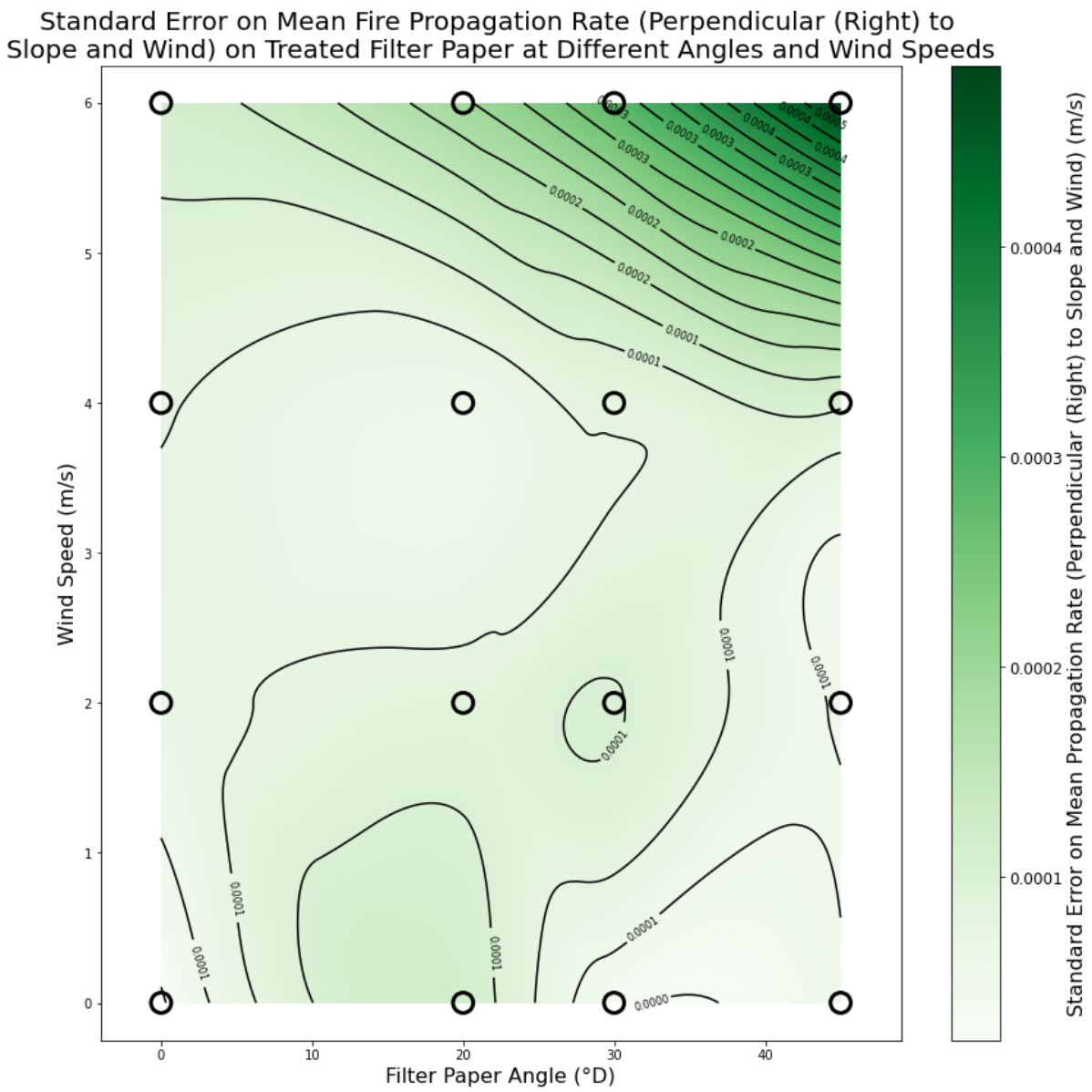

Figure 5.17: Standard Error on the mean propagation rate travelling perpendicular (right) to the slope and wind on treated filter paper. Different wind speed and angle combinations were examined. Black circles represent observed data points and cubic interpolation was used. Each observed data point had five samples excluding the data point with no slope and wind which had eight samples. Standard Error on mean propagation rate is denoted by darkness of colour and the contour lines. The highest standard error on mean propagation rate occurs near an angle of 45° with a wind speed of 6m/s. The lowest standard error on mean propagation rate occurs at an angle of 0° with a wind speed of 0m/s. Some fires extinguished near angles of 45° and wind speeds of 6m/s which may be why this area has a higher standard error.

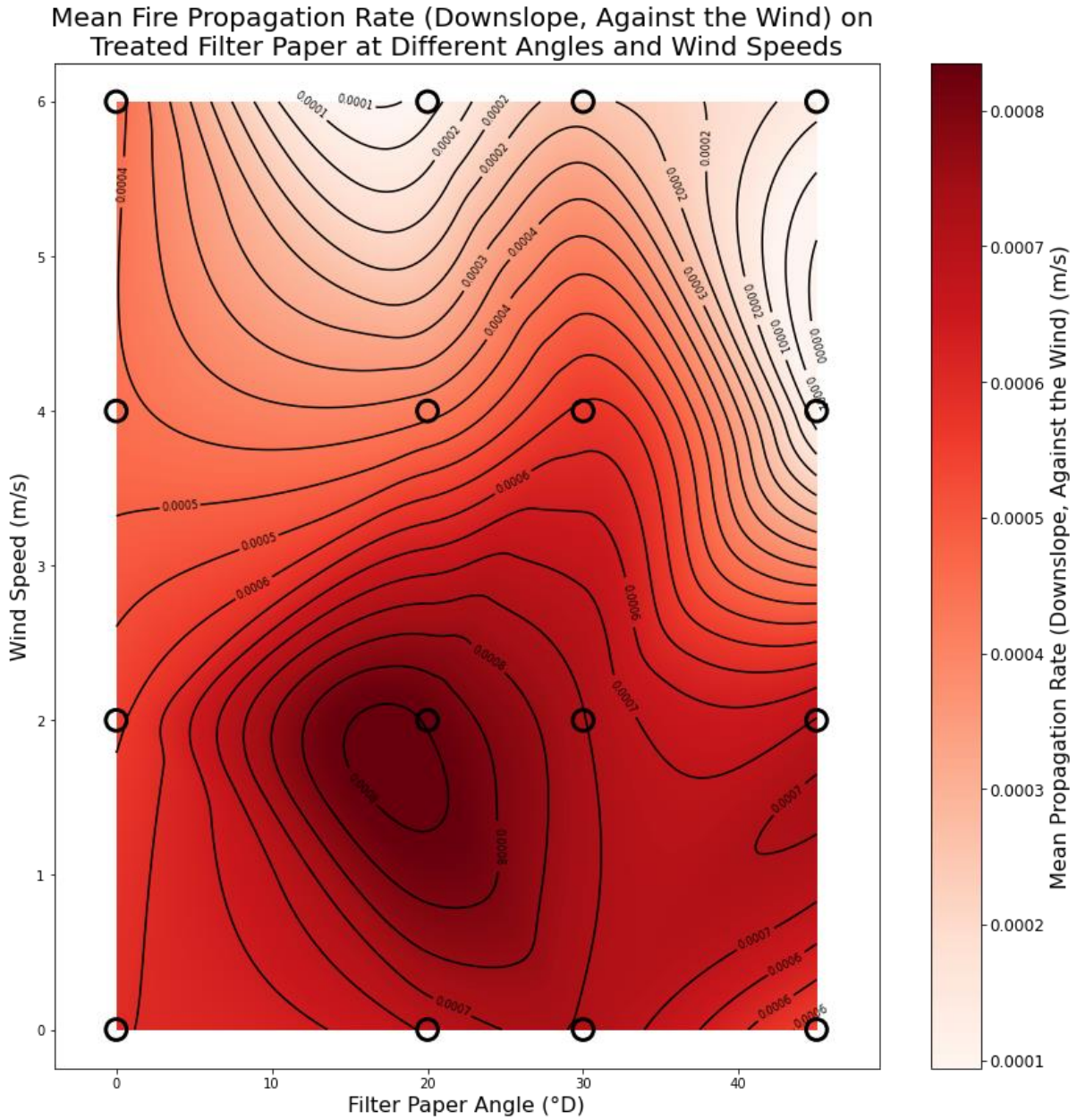

Figure 5.18: Mean propagation rate travelling downslope, against the wind on treated filter paper. Different wind speed and angle combinations were examined. Black circles represent observed data points and cubic interpolation was used. Each observed data point had five samples excluding the data point with no slope and wind which had eight samples. Mean propagation rate is denoted by darkness of colour and the contour lines. The fastest mean propagation rate occurs at an angle of 20° with a wind speed of 2m/s. The slowest mean propagation rate occurs at an angle of 45° with a wind speed of 4m/s. Some fires extinguished near angles of 45° and wind speeds of 6m/s. Fire travelling downslope and against the wind had the slowest propagation rates. These results also show a clear trend where wind speeds greater than 2m/s slowed down propagation rate.

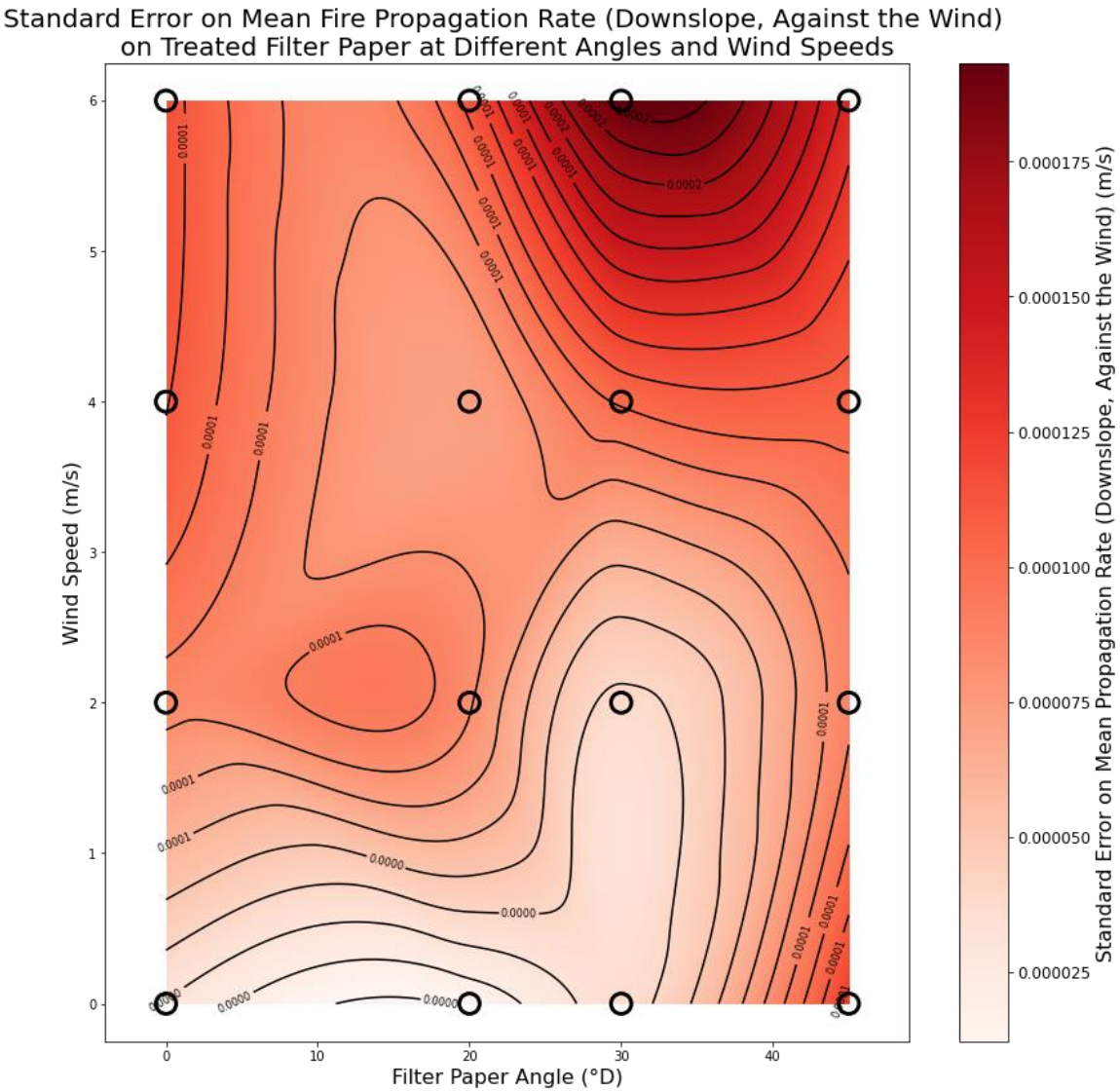

Figure 5.19: Standard Error on the mean propagation rate travelling downslope, against the wind on treated filter paper. Different wind speed and angle combinations were examined. Black circles represent observed data points and cubic interpolation was used. Each observed data point had five samples excluding the data point with no slope and wind which had eight samples. Standard Error on mean propagation rate is denoted by darkness of colour and the contour lines. The highest standard error on mean propagation rate occurs near an angle of 30° with a wind speed of 6m/s. The lowest standard error on mean propagation rate occurs at an angle of 20° with a wind speed of 0m/s. Some fires extinguished near angles of 45° and wind speeds of 6m/s which may be why this area has a higher standard error.

5.5 Comparison with a Wildfire Simulator

The experimental results were then compared to a wildfire simulator (Irregular Grid Software, IGS) which simulated wildfires using the Rothermel model [157] [46]. The Rothermel model is typically used to find the propagation rate of flaming fires while the experimental fires in this chapter are smouldering. An easy way to understand the Rothermel model is the area currently burning acts as a heat source, while fuel beds act as a

heat sink. The surplus heat produced will ignite the neighbouring area and influences the propagation rate. From an energy balance perspective, the underlying physics between a flaming fire and a smouldering fire are the same. Therefore, it was possible to compare the Rothermel model to the propagation rate of fire on the treated filter paper however, modifications were made to fit the Rothermel model to smouldering fires (including scenarios with wind or slope) which are explained below. While smouldering fire models exist [187], the Rothermel model was selected due to it being an industry standard wildfire model and as it was already implemented in the IGS.

The Rothermel model is normally presented using United States customary units. All other calculations in this chapter use SI units. Any calculations within the Rothermel model therefore required conversion (for formatting, when United States customary units are used, they will be placed in brackets next to SI units). This experiment used an updated version of the original Rothermel model where the net fuel load was changed from ((Eq. 2.8) to (Eq. 5.3)) and values used to calculate the optimum reaction velocity were changed from ((Eq. 2.11) to (Eq. 5.4)) [149], as seen below:

$$w_n = w_0(1 - S_T),$$

(Eq. 5.3)

$$A = 133\sigma^{-0.8189},$$

(Eq. 5.4)

where w_n is the net fuel load in kg/m^2 (lb/ft^2), w_0 is the oven-dry fuel load in kg/m^2 (lb/ft^2), S_T is the ratio of total mineral content in the oven-dry fuel load, A is an arbitrary variable used to find the optimum reaction velocity (Eq. 2.10) and σ is the surface area to volume ratio m^{-1} (ft^{-1}).

The Rothermel model requires a set of input parameters. These include the fuel particle density, fuel moisture content ratio, fuel depth, fuel surface area to volume ratio, fuel load, fuel low heat content and the total fuel mineral content. The values of these variables were sourced and calculated. The moisture of extinction was set to 0.3 as seen in many wildfire simulators [157] [66]. A fuel particle density of 1,590kg/m^3 was selected from the midrange density of microcrystalline cellulose [188] and a fuel moisture content ratio of 0.045 was set

as it is the midrange value for microcrystalline cellulose [189]. The values for microcrystalline cellulose were chosen as a substitute for the lack of available data for filter paper. Microcrystalline cellulose represents a pure form of cellulose. This is like filter paper, as seen by its low ash content [190] [172].

The fuel depth was estimated by adding the mean thickness of the filter paper to an additional fuel depth. The mean thickness of the filter paper was 1.45×10^{-4}m, this was measured using a micrometre. In the Rothermel model the density of fuel in the fuel bed is calculated using the fuel depth and other input parameters. The density of fuel determines if enough air which contains O_2 can reach the fuel particles, and the efficiency of heat transfer mechanisms which both influence the fire's propagation rate. Therefore, the oxygen release from the decomposition reaction of KNO_3 was accounted for by adding additional depth to the fuel. The mean difference in filter paper mass before and after being treated with KNO_3 was 4.112×10^{-4}kg. This was multiplied by 0.16, the mass ratio of O_2 released by the decomposition of KNO_3, to find the mean mass of O_2 per treated filter paper; this was 6.5×10^{-5}kg. The density of air is 1.204kg/m^3 and this was multiplied by 0.21, the ratio of O_2 in air to find the density of O_2 in air which is 0.253kg/m^3. The mean mass of O_2 per treated filter paper was then divided by the density of O_2 in air to find a volume of air containing the same mass of O_2 as the mean mass of O_2 per treated filter paper. This produced a value of 2.57×10^{-4}m^3. The additional volume of air was divided by 1.767×10^{-2}m^2, the area of the top surface (side facing the camera) of the filter paper to find the additional depth of the filter paper required to contain that volume of air which was 1.46×10^{-2}m. This was added to the thickness of the filter paper to get a total fuel depth of 1.47×10^{-2}m. The additional fuel depth represents approximately 99% of the total fuel depth.

The surface area to volume ratio was calculated using the surface area and volume of the filter paper. It measures the potential surface area of fuel that can ignite compared to the total volume of fuel, where its units are the reciprocal of length. This surface area was 1.767×10^{-2}m^2 which represents the top area of the filter paper. The volume of the filter paper, 2.56×10^{-6}m^3, was found by multiplying the area of the top side of the filter paper by

the filter paper thickness. The surface area to volume ratio was found by dividing the surface area by the volume to produce, 13,793.1m^{-1}.

The fuel load was found by calculating the mean mass of the treated filter paper and dividing by the area on the top side of the filter paper to produce, 0.115kg/m^2. The low heat content (net calorific value (NCV)) of the fuel was set as 22,980,000J/kg. This value was soured from data on packing paper in a dry state [191].

The ratio of total mineral content was calculated in two parts, the original mean ash content (minerals) of the untreated filter paper and additional mineral content which was added to account for KNO_2 released from the decomposition of KNO_3 during combustion. The ratio of ash content in untreated filter papers was 0.00125 [192]. This was multiplied by the mean mass of the untreated filter paper to find the mean mass of ash content in filter paper, $2.025x10^{-6}$kg. The following steps calculated the additional mineral content to account for KNO_2. The mean difference in filter paper mass before and after being treated with KNO_3 was $4.112x10^{-4}$kg. This was multiplied by 0.84, the mass ratio of KNO_2 released from decomposition of KNO_3 to find the mean mass of KNO_2 per treated filter paper which was $3.46x10^{-4}$kg. This was added to the mean ash content of the filter paper to get a mean total mineral content of $3.48x10^{-4}$kg per filter paper, which was a ratio of 0.171 for the mean treated filter paper mass.

The mean propagation rate (with one standard error) of the fire from the experimental data with no slope or wind was $6.163x10^{-4}$m/s (SE: $9x10^{-6}$m/s). The rate of spread outputted by the Rothermel model with fuels values adjusted for filter paper was $7.503x10^{-4}$m/s.

The Rothermel model was adjusted to model fire on treated filter paper in the direction of the wind. In Rothermel's original paper the measured wind speed was taken 10m above the ground. Wind speed was multiplied by a factor of 0.4 to estimate the actual wind speed at mid flame height [193]. This study also multiplied the wind speed by 0.4 as it was not possible to get a direct measurement on the surface of the paper. This was due to the size of the anemometer. The original and revised wind speed limits were removed from the Rothermel model as they did not align with the experimental data (Figure 5.20). It is

possible that smouldering fires may sustain combustion at higher wind speeds. The experiments showed that fires with slopes of 45° and wind speeds of 6m/s tended to extinguish. Fires began to extinguish at slopes as low as 30° (with a wind speed of 6m/s) and wind speeds as low as 4m/s (at a slope of 45°). This became more common with an increase in slope angle or wind speed. A wind limit could possibly be developed to reflect this, but many additional experiments would be required to accurately model this.

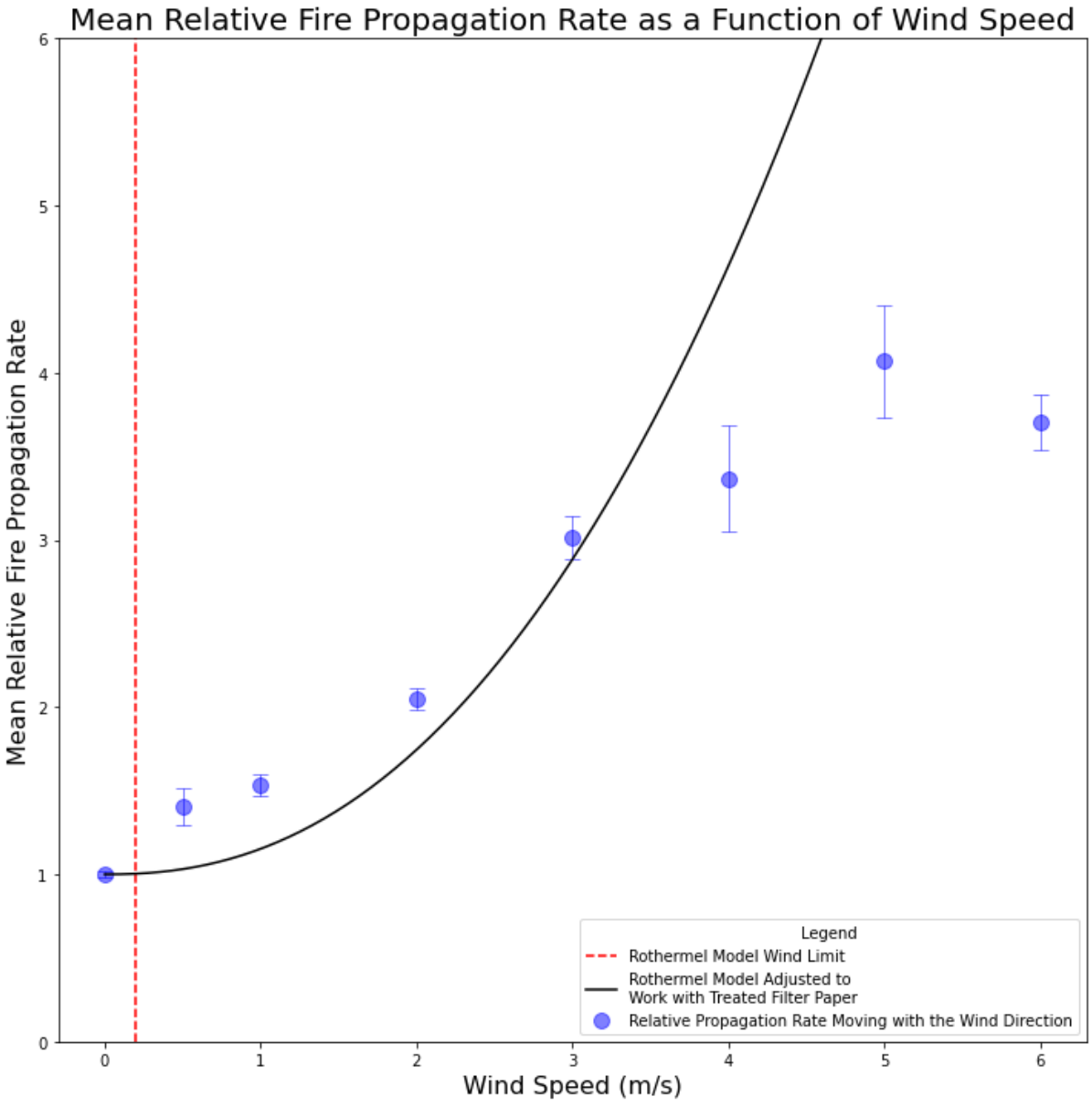

Figure 5.20: Comparison of the observed mean propagation rate with one standard error at different wind speeds with no slope, relative to the mean propagation rate with no wind. The red dashed line represents the wind limit, calculated using fuel properties from the treated filter papers. This wind limit was not used, and predictions were extrapolated above this limit. Observed fires started to extinguish at wind speeds of 6m/s where there was no slope.

The Rothermel model was adjusted to model fires on treated filter papers in the direction of upslope. It was adjusted by multiplying the slope factor by a coefficient found using manual binary search, resulting in the value of 0.03. This coefficient was used to give independent control of the slope factor. The original Rothermel model had no adjusting coefficient. The fitted Rothermel model may have required this coefficient due to the use of treated filter paper as fuel which produced smouldering fires and had a small fuel bed depth. This approach provided the best fit to the experimental results. It is also important to note that the Rothermel model measures rate of spread on a slope from a head on perspective not an aerial perspective (Figure 5.21) [194]. The original Rothermel model was designed to operate on slopes with a maximum angle of 45° as vegetation tends not to grow on slopes any steeper than this. Like the original Rothermel model the adjustments made to the model should only function up until 45°, after which the experimental data deviates from the slope factor (Figure 5.22).

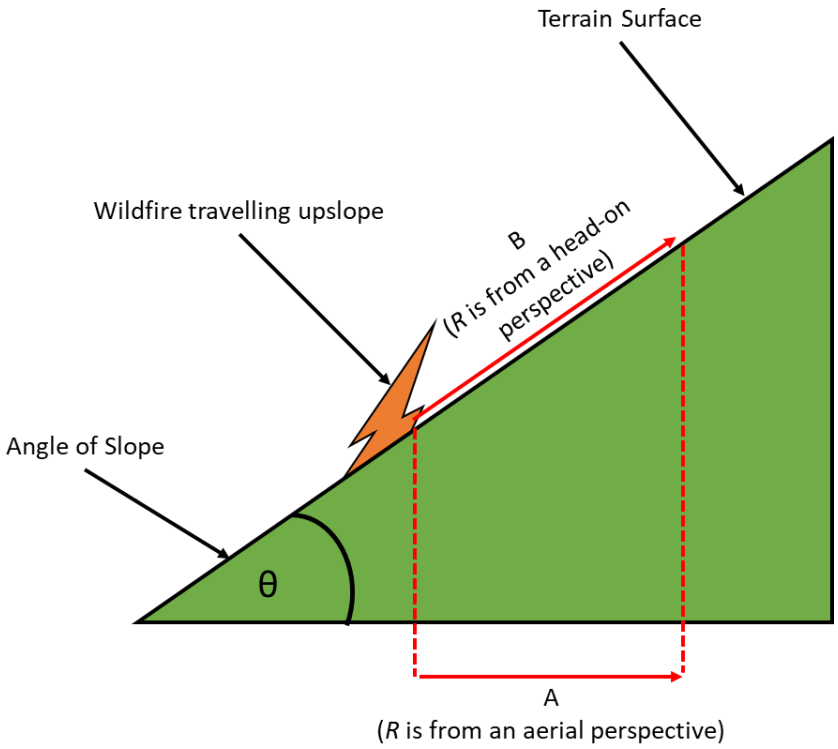

Figure 5.21: The two ways of interpreting the rate of spread (R) outputted from a wildfire model on sloped terrain. The distance A assumes R represents propagation from an aerial perspective while distance B represents propagation from a head-on perspective. Both perspectives are linked as $A = B(\cos(\theta))$. The Rothermel model outputs a value for R that uses the distance represented by B (the head-on perspective).

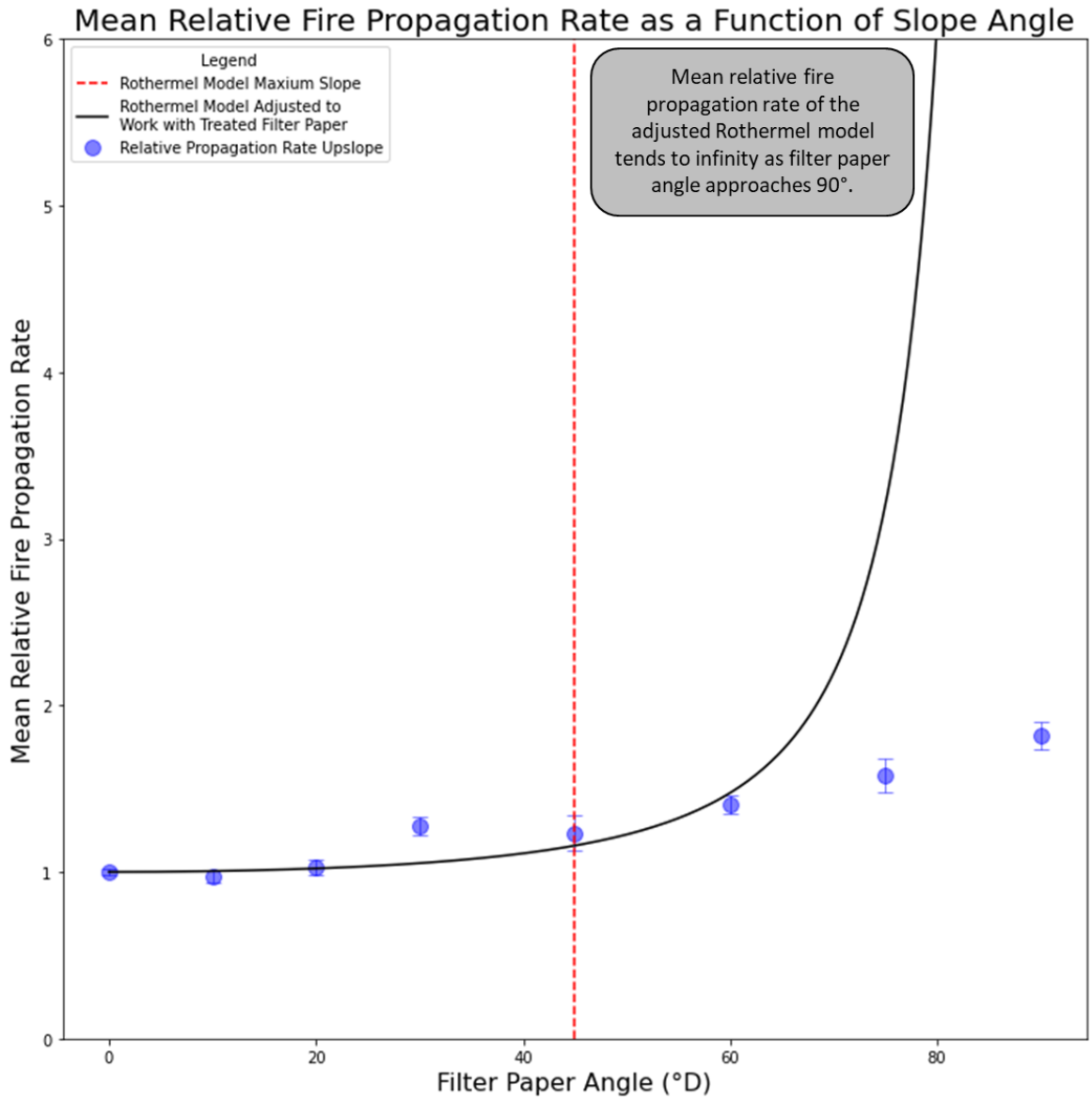

Figure 5.22: Comparison of the observed mean propagation rate with one standard error at different angles of slope with no wind, relative to the mean propagation rate with no slope. The red dashed line represents 45°, the Rothermel model's maximum slope.

A comparison was also made between the mean perimeter (outline) of the treated filter paper's burnt area as the fire propagated from the experimental results and the fire line representing the outline of the burnt area produced by the IGS (using the altered Rothermel model). For this chapter the IGS grid had 2,000 sites and simulations had a time step of 1 second. Wind speed and angle of slope were selected and all experimental burns that occurred in these conditions were used to compute the mean perimeter (the IGS also used these conditions to model the wildfire). The mean perimeter was computed by finding the mean distance from the ignition point to the edge of the burnt area at 10° intervals (by

using computer vision to rotate the image, get the bounding box of the contour, and measure distances). These distances were then used to place vertices at 10° intervals around the ignition point. Lines could then be drawn connecting each vertex to the vertices at the previous and next 10° interval. This crated a mean perimeter (Figure 5.23). The burnt area from the experimental data and the IGS were compared using a threat score as a metric [157]. The threat score calculates similarity of the two burnt areas by measuring the overlap area. This score is normalised to produce a value in the range of 0 to 1 (ranging from no overlap to complete similarity respectively) (all threat score weights were set to 1). This was done for each second of the video. A small number of frames were dropped when the computer vision algorithm could not evaluate the perimeter (Figure 5.24).

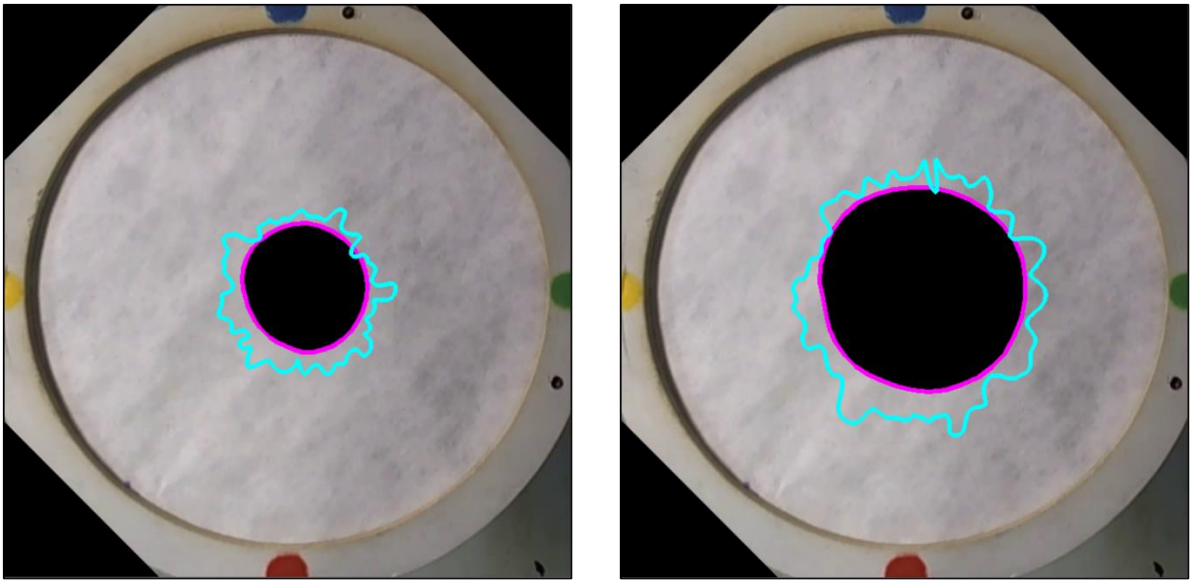

Figure 5.23: Comparison between the burnt area perimeters of the mean experimental results and the IGS with no wind and slope. The cyan line represents the perimeter of the IGS estimated burn area. The magenta line surrounding the black area is the mean perimeter (at 10° intervals around the ignition point) of the eight burns with no slope and no wind. The data is presented on top of a still image of filter paper where no burning had occurred. The left image is from 25 seconds after initial ignition and the right image is from 50 seconds after initial ignition. The IGS, estimated burn area modulates due to the IGS propagating fire on a grid instead of continuous space.

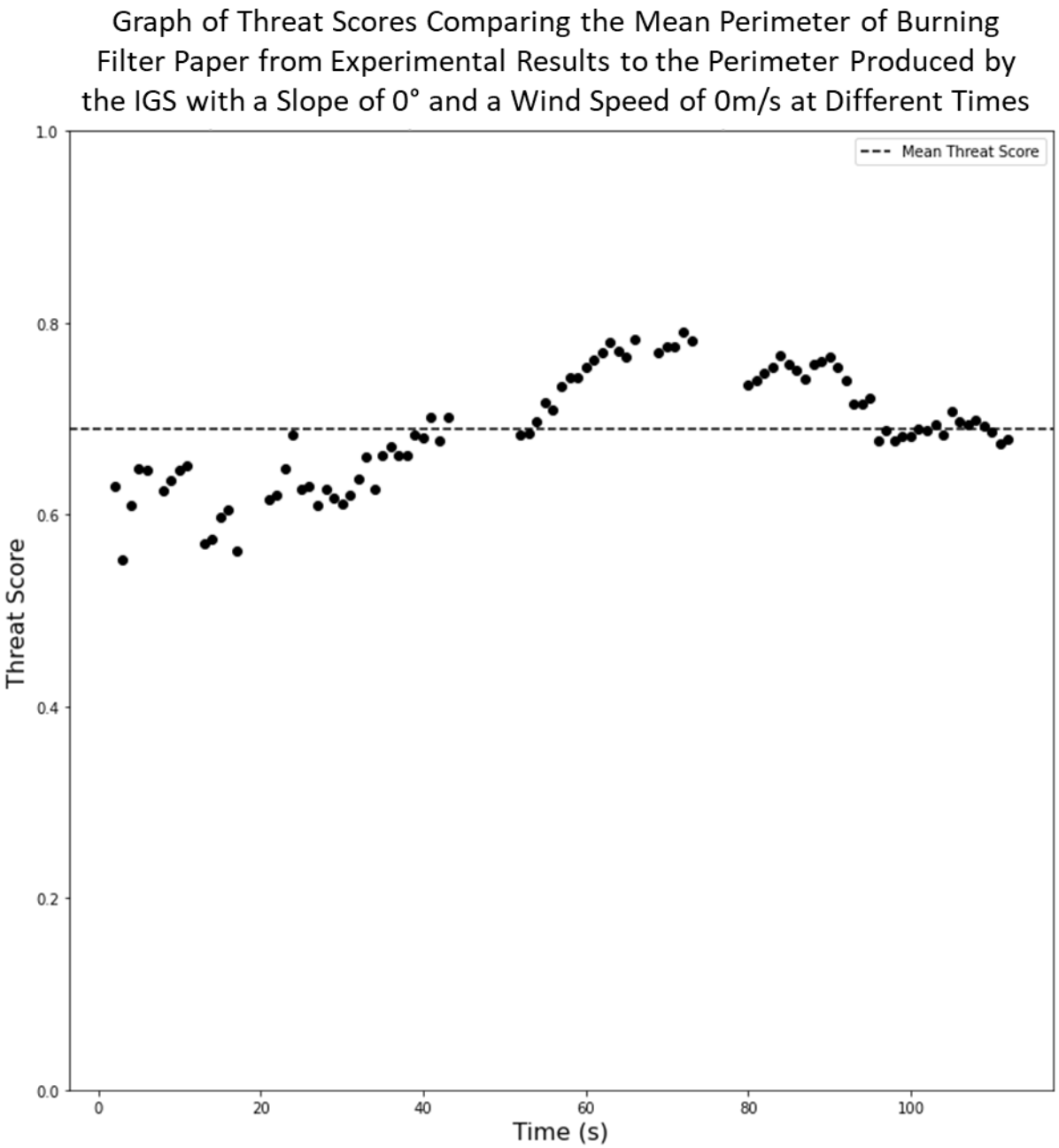

Figure 5.24: The threat score (used to measure similarity) comparing the burnt area perimeters from the mean experimental data and the IGS at each second of the recorded video. A dashed line is used to denote the mean threat score. Any invalid threat score values were ignored, visible by a gap on the x-axis between graphed measurements. Values were recorded from initial ignition until the burnt area reached the boundary.

5.6 Discussion

In this section the propagation rate of fire on treated filter paper at different angles of slope and wind speeds was examined. The experimental data showed fire propagation rate increased in the direction of the wind and upslope. The opposite was true against the

direction of the wind and downslope. There was a slight increase in mean propagation rates for the fire spreading perpendicular to the wind. This may be due to an increased amount of oxygen reaching the fire, speeding up combustion (like blowing on a campfire) (Figure 5.10).

There was variance in the propagation rate seen in in the comparison of different wind speeds and angles of slope (Figure 5.10) (Figure 5.11). There could be many reasons for this variance including differences in the layout of microscopic paper fibres or minor variations in the concentration of KNO_3 within each treated filter paper. The variance of propagation rates could also be impacted by other factors such as the wind extinguishing the fire at high wind speeds. The standard error of propagation rates could be reduced by increasing the sample size.

Burning treated filter paper at an angle of 30° and a wind speed of 6m/s produced the fastest mean fire propagation rate. This may be due to the wind delivering heat energy from combustion into treated filter paper that is upslope from the area which is burning. Any fires that extinguished resulted in slower mean propagation rates.

The Rothermel model fitted to the experimental data exhibited similar propagation behaviour to what was observed. This was supported by the threat score which quantified the similarity between the fire lines of the mean experimental data and the IGS running the fitted Rothermel model, however further studies could be performed to validate this. The threat score had a mean value 0.69 across the burn duration, for the tested burn. It is important to note that the Rothermel model used in this chapter was fitted to work with smouldering fires on treated filter paper. The original Rothermel model would not have as high of a threat score. Future work could include comparing the fitted Rothermel model to other smouldering models.

The effect that angle of slope has on the propagation rate of burning paper has been examined in the past [75]. Other studies have also experimented with tracking the propagation rate of fire on paper treated with KNO_3 [106] [77]. From a literature review the research presented in this chapter appears to be the first to record the effect wind has on

fire propagation on treated filter paper. It also appears to be the first to compare experimental data from burning paper to a modified version of the Rothermel wildfire model or a framework for modelling wildfires such as the IGS.

5.7 Conclusions

A versatile apparatus was constructed to perform burning experiments on treated filter paper. These experiments investigated how slope and wind effect the propagation of fire across the paper. The propagation rates were compared to a modified version of the Rothermel model. Test burns can be completed relatively quickly compared to real wildfires or fuel beds comprising of forest litter. Variables such as fuel, wind or slope can be changed and the rate of spread can be recorded quickly, providing experimental results for the effects these variables have on the rate of spread. An extension to this research could be useful for constructing computational wildfire models supported by a large amount of experimental data.

The approach made effective use of a novel computer vision algorithm to process large datasets efficiently. All 123 video sequences (125,666 frames in total) were processed in approximately 35 minutes in a single run on an Intel® Core™ i7-10870H Processor @ 2.20GHz. The algorithm and values used in the computer vision pipeline were optimised so no interventions were required. The developed computer vision pipeline means that even larger sets of videos could be processed relatively quickly compared to manually measuring the distance fire had spread at different times while burning. The use of an ideal overhead view also means that distances measured by this algorithm should be more accurate than those measured using raw unprocessed video data.

The results of this experiment have utility for demonstration and educational purposes. In relation to wildfires, it allows a large parameter space to be explored with simple equipment and a small quantity of consumables. There are limitations to this work as the size of the filter paper prevents research into fires which burn over larger areas and for longer durations. Scaling up a fire from a filter paper to a wildfire may also be prone to errors due

to the large difference in area between them. The differences between flaming and smouldering fires would also have to be carefully considered.

The research has opened many avenues for future work. A modification to the apparatus to allow multiple ignition points could assist research on techniques to locate the ignition points from the fire line [195]. A demonstration on the effects of moisture on burn rate could also be completed. If the position of moisture (or KNO_3) could be controlled over the surface of the paper, then it may be possible to paint a map of a region of interest and simulate a wildfire in that region. Other oxidising agents could also be used to treat the filter paper such as potassium chlorate or potassium permanganate; however, the safety requirement would differ from KNO_3.

Overall, this experiment has shown that burning filter paper in a desktop environment has similar properties to a scaled down wildfire. A computer vision algorithm made it possible to track the burned area of a progressing fire. The desktop approach combined with the automated data analysis system provides a means of collecting a large dataset. Fire travelled uphill and in the direction of the wind the quickest. The results were compared to a wildfire simulator using a fitted version of the Rothermel model and they both produced similar burn scars. This will allow access to many control parameters affecting burn rate to be explored (which includes wind speed, slope, moisture, fuel-type, ignition points and interventions).

Chapter 6 Applying Wildfire Modelling to the Real World with DecaMap

The research discussed in this thesis was developed in cooperation with a team called DecaMap. They are working on developing a platform to assist emergency services.

DecaMap's platform uses real-time intelligence for more effective and safer emergency event management, protecting citizens, safeguarding responders, minimising damage to property and environmental impacts [43]. DecaMap was originally called CoPilot-AI and were formed to compete in the SFI-Defence Organisation Innovation Challenge 1 [83]. This challenge was to enhance the efficiency and effectiveness of the fire extinguishing capability of rotary-wing aircraft.

DecaMap have collaborated with several stakeholders that have an interest in preventing and suppressing wildfires such as the Irish Fire Service, Irish Air Corps, Coillte, National Parks and Wildfire Service, Civil Defence, An Garda Síochána and the Health Service Executive.

6.1 DecaMap Technology

The main user interface of DecaMap's platform provides access to a Common Operational Picture (COP). The COP allows users to access and interact with map layers and video streams. The system also provides a communication link so useful information regarding an event can be send and received by users. As an example, using the COP a chief fire officer working from a Fire Command Support Unit could notify firefighters working in the field that are managing a wildfire about the location of nearby civilians that may be in danger.

An integral part of DecaMap was ensuring that communication using the COP would function in rural areas as that is where a majority of wildfire tend to occur. This can sometimes be challenging as mobile data tend not to be as readily available in rural areas [196]. Therefore, a device was developed that allowed users to connect to the internet using 4G, 5G and Satellite internet simultaneously. The device contained Subscriber Identity

Module (SIM) cards for various wireless network operators in Ireland. The ability to connect to multiple mobile network operators and Satellite internet simultaneously ensured that in most rural areas the internet would be accessible and therefore data could be sent a received using the COP.

The COP provided access to Geocoded Panoramas at Altitude (PanAlt) of the area which was recorded from a stationary drone. The PanAlt was a panorama image taken from a high altitude over the area of interest. The PanAlt panorama image was interactable where users were able to place markers by clicking on a position on the image, which would correspond to that same position on the COP map. Distances could also be measured between these markers. The PanAlt gave chief fire officers an improved perspective of the terrain than would be visible from the ground, which could provide them more information when preparing to deploy firefighters.

The COP also provided access to real time Aerial Drone Video (ADV) which members of the fire service could move for a better perspective on the terrain. ADV was recorded from a moving drone where the position of the drone and its field of view were both recorded on the COP map. As of writing it is not currently legal in Ireland to fly a drone near other aircraft, therefore use of ADV was restricted to times when there were no firefighting aircraft operating in the sky [197]. Work is advancing to allow tethered drones to fly in such regions.

Geocoded Machine Learning (Geo-ML) was performed on drone footage. Geo-ML used both coloured RGB cameras and thermal cameras attached to a drone as input. The frames from videos recorded on these cameras were processed using an object detection, machine learning algorithm to classify, locate and segment the position of an object of interest in a frame. Geo-ML was trained to detect people, animals, buildings, vehicles, and fire. The locations of these objects were then plotted to their corresponding position on the COP map. Geo-ML allowed fire officers to see what objects of importance were near a wildfire so they could plan accordingly.

The work from this thesis that contributed towards COP was focused on providing real time visualization of wildfire progression. The software developed could also be used for planning. The IGS and a dynamic fire line simulator were used to model wildfires in the COP. The boundary of the burn scars from these wildfire modelling frameworks were displayed on the COP map as isochrones (a line representing the location of the fire line at a certain time). This map layer could be used by professionals who manage wildfires to forecast the wildfire's behaviour and plan interventions.

6.2 Dynamic Fire Line Simulator

The dynamic fire line simulator was an interactive piece of software developed by Charles Markham and Conor Hackett. It modelled wildfire propagation on a hexagonal gird. Each cell in the grid had fuel and elevation data associated with it, sourced from Copernicus [97]. Wind data such as direction, speed and spread were set by the user. The ignition point of the wildfire was also set by the user.

Wildfire propagated from any cell that was currently on fire to its neighbouring cells containing vegetation. The dynamic fire line simulator took a statistical approach to propagation where the wildfire spread based on random chance to different neighbours which is different to the semi-empirical approach seen in Chapter 3. The probability of a wildfire propagating to a neighbour was determined from the type of land cover, slope and wind. Data for the propagation rates on different types of land cover were sourced [40] and the probabilities were calibrated so that on average the propagation rate agreed with the literature. Existing literature states that for every 10° increase of an angle in slope a wildfire's propagation speed approximately doubles [198], this is widely known as a rule of thumb amongst members of the Irish Fire Service. This closely resembles the effect slopes up to 45° have on propagation rate in Rothermel's estimation. It also parallels the results seen in Chapter 5 before the slope factor has been multiplied by the coefficient 0.03 which was found in that chapter. The effect wind had on wildfire propagation in the dynamic fire line simulator was manually set by the user.

The primary attraction of the dynamic fire line simulator was the ability for a user to add interventions in real time to any location they choose. There were three types of interventions: fire beaters, fire breaks and water which are the same forms of interventions used by professionals that manage wildfires in Ireland.

Using the dynamic fire line simulator, it was possible to see information about any cell such as the elevation, slope direction and angle, type of land cover, rate of spread for that land cover, position of that cell in different coordinate systems and the nearest Eircode by hovering over it. When the simulated wildfire propagated on terrain, data about the wildfire were shown to the user. The data included the current area of the wildfire, the number of properties denoted by Eircodes destroyed and the current number of individual fire beaters, required to contain the fire along all user placed fire beater intervention lines (Figure 6.1).

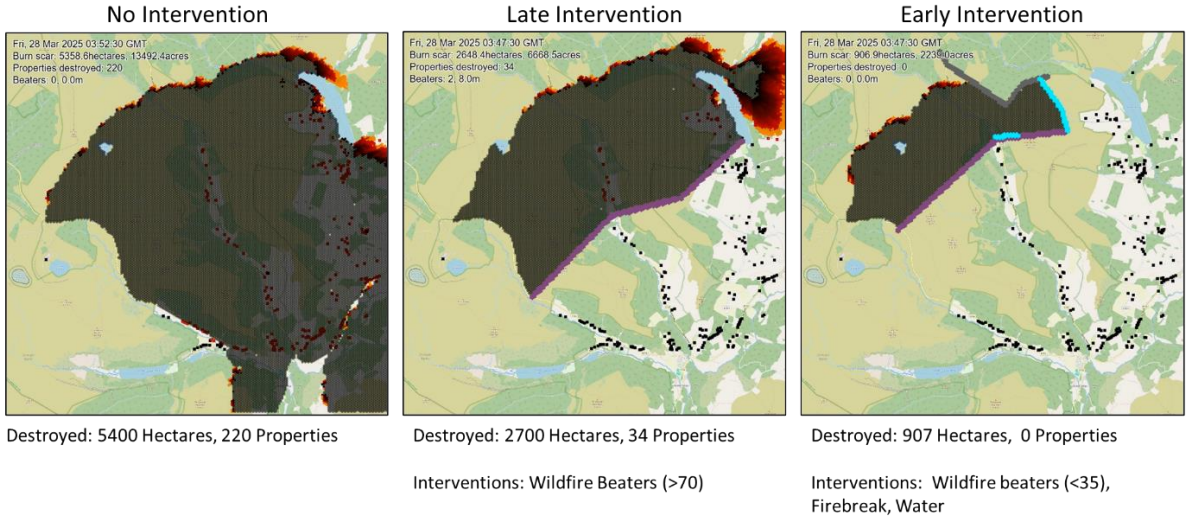

Figure 6.1: Dynamic fire line simulator used to model three possible intervention scenarios chosen for the EU MODEX exercise.

Stakeholders that manage wildfires shared their experiences on wildfire behaviour and gave advice that informed the design of the software. Members of the Irish Fire Service provided scenarios that they would like to be simulated, requested Eircode tagging, a property damage counter and a live feed displaying the number of fire beaters required. Coillte suggested adding fire breaks as a form of intervention. The Irish Air Corps wanted bigger

buttons to make inputs easier. The National Parks and Wildlife Service were interested in recovery after the wildfire and burn scars.

6.3 DecaMap Exercises

DecaMap was involved in multiple events including training exercises during development of the COP. These events were documented as they informed the direction of the research within this thesis.

The first exercise was called the CoPilot-AI Exercise FireGuard and took place on the 06/10/2022 in Lough Dan, Co. Wicklow, Ireland. The DecaMap team (Maynooth University, National Centre for Geocomputation), along with the Wicklow Fire Service, Irish Air Crops, Coillte, National Parks and Wildlife Service, An Garda Síochána and Microsoft were all in attendance at this event (Figure 6.2). The FireGuard exercise was the culmination of the work done for the SFI Defence Challenge at that point. The aim was to demonstrate the functionality of the main components of DecaMap in a live event. During the FireGuard exercise ForeFire and the dynamic fire line simulator were used to simulate where a wildfire would spread from three ignition points in Lough Dan (Figure 6.3) (Figure 6.4) (Figure 6.5).

Figure 6.2: DecaMap team and collaborators at FireGuard exercise.

Figure 6.3: Conor Hackett testing the dynamic fire line simulator during the FireGuard exercise.

Figure 6.4: Left: Dynamic fire line simulator modelling a wildfire during the Fire Guard exercise at different time steps (advancing in time from top to bottom). Right: An Irish Air Corps helicopter dropping water from a filled Bambi-Bucket on flares representing the wildfire being simulated.

Figure 6.5: Charles Markham (left) and Conor Hackett (right) facing the camera in front of an Irish Air Crops helicopter during the FireGuard exercise.

DecaMap also competed alongside a group in Microsoft to partially integrate DecaMap into Microsoft Teams as part of Microsoft's Fix.Hack.Learn 2023 hackathon (Figure 6.6). This event took place in Microsoft, Co. Dublin, Ireland from the 13/02/2023 to the 15/02/2023. During the hackathon the dynamic fire line simulator was integrated into Microsoft Teams and buttons were added so the program could interpret user input from a touchscreen instead of keyboard (Figure 6.7). The team at Microsoft which DecaMap collaborated with on this project won first place in the international hackathon [199].

Figure 6.6: DecaMap team and collaborators with the Microsoft team during the Fix.Hack.Learn 2023 hackathon.

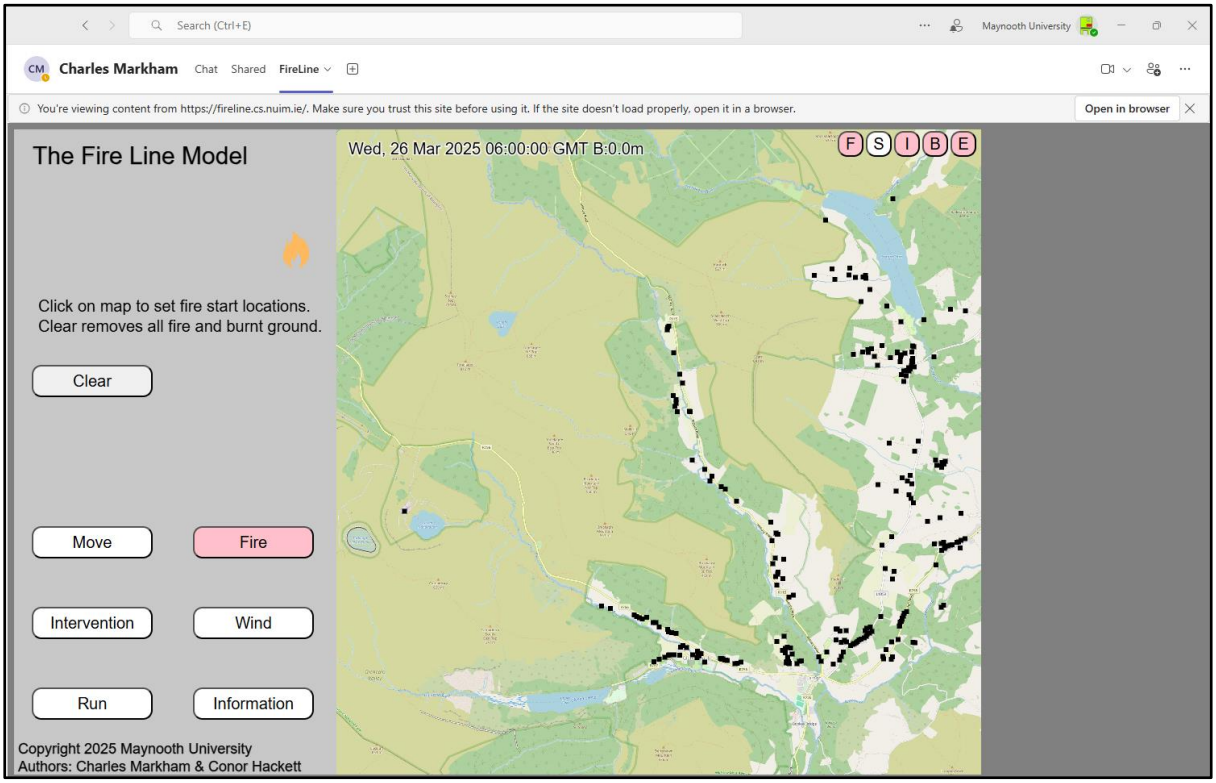

Figure 6.7: The dynamic fire line simulator integrated into a Microsoft Teams environment.

The SFI-Defence Organisation Innovation Challenge 1 Prize Event to announce the winner of the challenge took place on the 22/02/23 at Casement Aerodrome (Baldonnel), Co. Dublin Ireland. Twenty teams participated in the challenge. The dynamic fire line model was displayed again for interested parties (Figure 6.8). The Tánaiste, Minister for Further and Higher Education, Research, Innovation and Science with the Director General of SFI (RI) presented the DecaMap team with the winning prize to continue developing DecaMap (Figure 6.9) [200].

Figure 6.8: Conor Hackett in front of the dynamic fire line simulator at the SFI-Defence Organisation Innovation Challenge 1 Prize Event.

Figure 6.9: DecaMap team and collaborators at the SFI-Defence Organisation Innovation Challenge 1 Prize Event.

On the 25/10/23 the DecaMap Realtime Aera Surveillance & Mapping (RASM) Wildfire Exercise took place at Maynooth University. DecaMap displayed the updated tools to stakeholders online. These included the Irish Fire Service, Irish Air Corps, drone service providers and manufacturers. During RASM the IGS was shown integrated into the COP, and the dynamic fire line simulator was shown separately later (Figure 6.10).

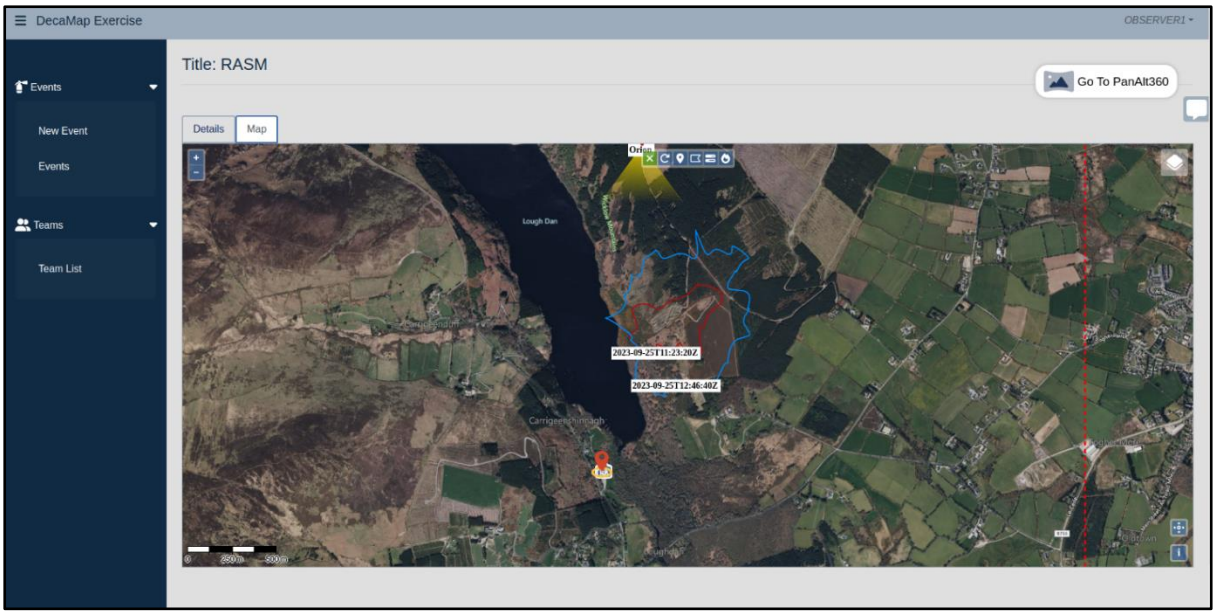

Figure 6.10: IGS integrated into the COP during the RASM event.

DecaMap were also involved in the EU Module Exercises (MODEX) event where firefighters from the EU came to further develop an EU wildfire fighting strategy (Figure 6.11) [201]. This event took place on the Wicklow Mountains at the 26/03/25 [202]. During the event DecaMap showed how the COP operates to firefighters from Ireland and the EU. The dynamic fire line simulator was used and shown to firefighters where its output was also displayed in the COP (Figure 6.12) (Figure 6.13). During the EU MODEX event the dynamic fire line simulator ran three possible scenarios where a wildfire was started on Turlough Hill, Co. Wicklow, Ireland. The three scenarios consisted of no interventions, a late intervention and an early intervention (Figure 6.1). The timings and position of personnel during the exercise were guided by the simulation. The position of the fire beaters in the early intervention simulation was the same as the firefighters in the real exercise (Figure 6.14).

Figure 6.11: Charles Markham (left) and Conor Hackett (right) in front of the Fire Command Support Unit where the dynamic fire line simulator was shown at EU MODEX.

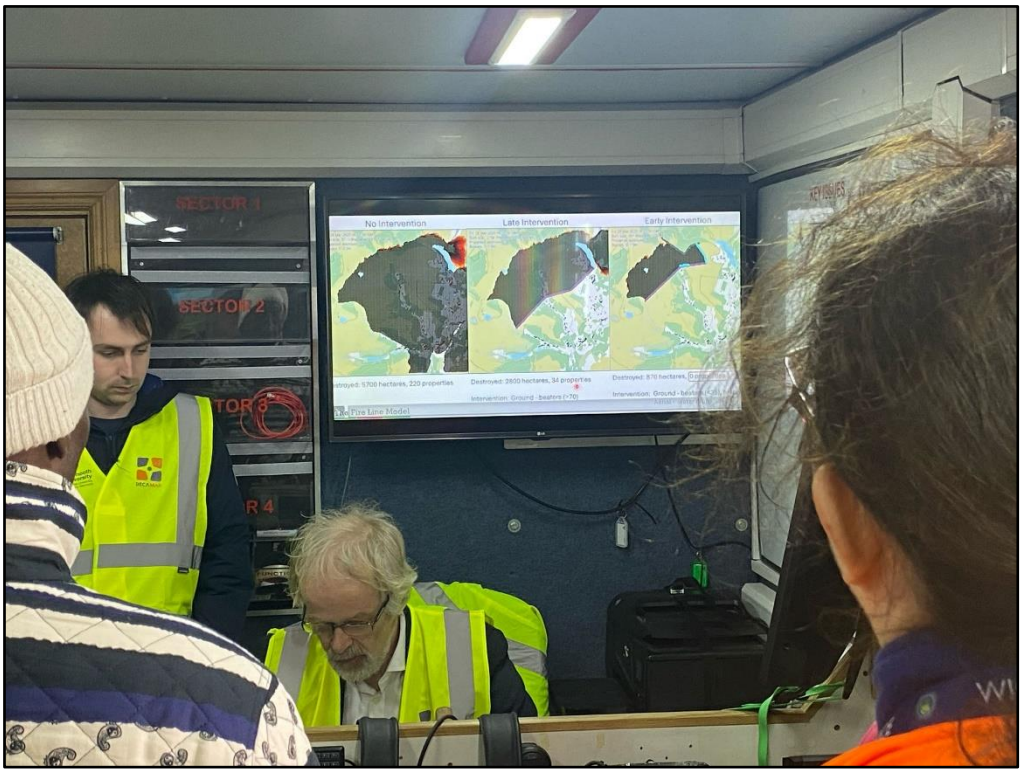

Figure 6.12: Conor Hackett (left) and Charles Markham (right) demonstrating the functionality of the dynamic fire line simulator to EU firefighters for EU MODEX.

Figure 6.13: Estimated fire lines displayed in COP from the dynamic fire line simulator for EU MODEX.

Figure 6.14: Fire beaters at the intended fire beater intervention line selected in the dynamic fire line simulator for EU MODEX (Image credit: RTÉ).

Chapter 7 Conclusions

This chapter discusses the innovation that contributed to the contents of this thesis, the artefacts that were created in this process and the potential for future work which could continue from the research in this thesis.

7.1 A Reflection of Work Completed

This thesis described novel approaches to many areas of wildfire modelling.

The first area described developed software called the IGS which used a grid generated from a Voronoi diagram to simulate wildfires using the Rothermel model. The performance of differently formatted grids containing regular and irregular shapes was tested. Regular grids simulated wildfires the fastest while irregular focused grids with higher resolutions over areas of interest produced results of the highest similarity when compared to ForeFire. This research showed that grid-based propagation can potentially produce similar but more computationally efficient results than continuous propagation frameworks.

The IGS created a platform that helped progress other areas of wildfire research such as the Wildfire Source Genetic Algorithm. The WSGA estimated the ignition points and propagation durations of wildfires given their burn scar. The WSGA was a novel concept in the field of wildfire modelling as no literature with the goal of automating the detection of wildfire ignition points using burn scars was found.

A physical desktop model was also develop using treated filter paper as a fuel source. It was possible to measure the effects both wind and slope had on the fire propagation rate using this apparatus. Propagation rate was measured using a traditional computer vision pipeline. The IGS running a modified version of the Rothermel model was also compared to burns performed using the physical desktop model.

The research in this thesis had practical applications. Experts such as members of the fire services and other interested parties provided feedback and advice on certain research

topics. The software developed in this thesis was also used during multiple wildfire training exercises.

Overall wildfire modelling is useful for predicting how wildfires will propagate. Wildfire modelling can provide valuable information to groups such as the Fire Service. Typically, wildfires are supressed by these groups to prevent damage to infrastructure, the environment and the ecosystem along with protecting lives. However, it is important to note that wildfires cannot just be considered a natural disaster, as the effects of wildfires in an ecological analysis are not always negative [35]. The impact of some wildfires can be small and possible to control. Lager wildfires can be beyond management, but simulations may provide advice on evacuation [203]. An important aspect of simulation is to help manage risk and prevent wildfires before they begin.

7.2 Thesis Artefacts

The following is a list of artefacts from the research journey.

7.2.1 Publications

This section includes publications emanating from work completed.

Journal Publications

- An Efficient Method to Simulate Wildfire Propagation Using Irregular Grids published on the 28/02/2025 in Natural Hazards and Earth System Sciences [157].
- A Method to Identify Wildfire Ignition Points and Propagation Durations from Burn Scars Using Genetic Algorithms accepted for publication on the 17/12/2025 in Fire Technology [195].
- Wildfire Modelling using a Desktop Apparatus submitted on the 19/12/2025 to Physics Education [204].

Conference Publications

- A Computer Vision Technique for the Validation of Wildfire Models using a Desktop Apparatus published on the 04/09/2025 at the Irish Machine Vision and Image Processing Conference 2025 [176].

Book Chapters

- Machine Vision Applied to Entomology published on the 01/02/2024 as part of Modelling Insect Populations in Agricultural Landscapes [113].

7.2.2 Presentations

- Voulez-vous voter avec moi ce soir: Exploring 2017 French Presidential Election Results (CRT Winter Symposium 2022) (Contained a description of software developed to perform simple land cover classification.)
- Voulez-vous voter avec moi ce soir: Exploring 2017 French Presidential Election Results (Maynooth Loave Data Week 2022) (Contained a description of software developed to perform simple land cover classification.)
- Fire Line Progression Modelling (Lough Dan, Wicklow Mountains)
- Fire Line Progression Modelling (Maynooth Research Week 2022)
- Comparison of Edge Computing Platforms for Object Classification and Localisation in Camera Traps (CS Postgraduate Poster Session 2022) (Contained a description of software that could be used for monitoring ecological recovery after a wildfire.)
- A Step Towards Automating Sensitive Data Flagging in the CSO (Maynooth Love Data Week 2023) (Contained a description of software designed to work efficiently with large quantities of data.)
- DecaMap: Fire Line Modelling (Microsoft Ireland)
- DecaMap: Fire Line Modelling (Casement Aerodrome)
- Essential Expense Tracker (European Commission Big Data Hackathon) (Contained a description of software designed to work efficiently with large quantities of data.)
- Using Edge Computing to Automate Insect Detection (CASI 2023) (Contained a description of software that could be used for monitoring ecological recovery after a wildfire.)
- Enterprise Alliance (EA) Placement Talk (CRT Foundations of Data Science I) (Contained a description of software designed to work efficiently with large quantities of data.)

- Development of a Physical Desktop Approach to Wildfire Modelling (CS Postgraduate Poster Session 2023)
- DecaMap: Fire Line Modelling (DecaMap RASM 'Live' Exercise, Maynooth University)
- DecaMap: Fire Line Modelling (Digital technologies to support climate action and wildfire response (Research Week 2023))
- Investigation of Wildfire Simulations using Irregular Grids (Hamilton Institute Student Seminar)
- Investigation of Wildfire Simulations using Irregular Grids (Kind Suggestions Workshop (Maynooth University))
- Finding Wildfire Sources (DecaMap Review)
- Wildfire Modelling Research Summary (Major Emergency Management Conference)
- A Method to Identify Wildfire Ignition Points and Propagation Durations Using Genetic Algorithms (CS Postgraduate Poster Session 2024)
- A Method to Identify Wildfire Ignition Points and Propagation Durations Using Genetic Algorithms (Hamilton Institute Research Week Talks 2024)
- Wildfire Modelling Research Summary (Chief Fire Officers' Association Technical Seminar 2024)
- A Method to Identify Wildfire Ignition Points and Propagation Durations Using Genetic Algorithms (Young-ISA 2024)
- Fire Line Modelling on a Dynamic Platform (EU MODEX)
- Wildfire Modelling using a Desktop Apparatus and Computer Vision (TSE Mini Symposium)
- DecaMap FireLine Prediction - Online Cloud-based App - Copernicus Services (Meeting with European Forest Fire Information System)
- Visit of Oklahoma State University (Drones/Geo-ML/Wildfire etc) (Meeting with Stakeholders from Oklahoma)
- A Computer Vision Technique for the Validation of Wildfire Models using a Desktop Apparatus (Irish Machine Vision and Image Processing Conference (IMVIP) 2025.
- A Method to Identify Wildfire Ignition Points and Propagation Durations from Burn Scars Using Genetic Algorithms (Joint Conference of the Italian and Eastern Mediterranean Regions of the International Biometric Society (IBS-IR-EMR2025))

- A Method to Identify Wildfire Ignition Points and Propagation Durations from Burn Scars Using Genetic Algorithms (Research Connections Faculty of Science & Engineering)
- Simulating Disease in Periods of Low Mobility Using a Hybrid Diffusion and Compartmental Model Built on Geographic Data (TSE Mini Symposium)

7.2.3 Software and Hardware Developed

- A forward running wildfire propagation simulator called the IGS.
- An inverse wildfire propagation tool called the WSGA.
- An interactable wildfire propagation simulator called the dynamic fire line simulator.
- An apparatus including an igniter and wind tunnel for burning and recording treated filter papers.
- A computer vision algorithm to track burning fire on treated filter paper.
- A program to analyse collected data from burning treated filter paper.
- A YOLOv5 trained model on edge computing for monitoring recovery of small ecosystems, commissioning experiment was detecting wasps (*Vespula vulgaris*).

7.2.4 Awards

- Research Ireland Centre for Research Training in Foundations of Data Science Three Minute Thesis People's Choice Award (2025).
- Joint Conference of the Italian and Eastern Mediterranean Regions of the International Biometric Society (IBS-IR-EMR2025) PhD Student Poster Award for outstanding contribution to the field of Biostatistics and Bioinformatics.

7.3 Future Work

Future work includes further developments of each program guided by the conclusions in Chapter 3, Chapter 4 and Chapter 5.

7.3.1 Increasing Speed

Improvements to computational times increases the number of applications for modelling. In the research presented within this thesis the increased speed led to two developments. These include genetic algorithms to estimate the location of ignition points and a dynamic interactable wildfire modelling framework with real time wildfire simulation. Further developments in performance would allow larger areas to be simulated quicker.

The IGS can simulate multiple wildfires in the same region very quickly. Many wildfires with different properties (excluding fuel and elevation) such as ignition point locations and wind properties could be sequentially simulated to explore a wide range of scenarios. This has potential utility for the Fire Service to assist with wildfire management planning.

The experimental results found using the desktop apparatus showed that wildfires could be modelled using paper. The use of a desktop apparatus allows burning of many fuel beds in different conditions to build physical models. This is difficult with larger fuel beds. The existing desktop apparatus could be used to explore the effect of KNO_3 and moisture on the fire propagation rate. A comparison of flaming and smouldering fires could also be made.

7.3.2 Additional Features

During this thesis professionals who manage wildfires provided interesting and useful insights in the world of wildfire management. Further developments in all areas of wildfire modelling could be informed with more detailed and longer discussion with them.

Members of the Irish Fire Service suggested smoke modelling as an area of interest due to the increased difficulty involved with fighting wildfires in the presence of smoke. Interest has also been expressed in evacuation modelling in the presence of fire; this could help inform members of the Fire Service how best to move groups of people out of danger.

A method to detect burn scars using satellite imagery could also be useful. There is a lack of opensource data showing a timeseries of wildfires propagating in Ireland. If in the future it

is possible to get more frequent satellite images over the same area in the Wicklow Mountains (e.g. with additional satellites), it could be feasible to compare computational wildfire models to a timeseries of actual wildfires. Drone imagery may also provide a means of collecting the data.

7.3.3 Quantifying Risk

Another large area of wildfire modelling that could be explored is wildfire risk modelling. These models do not simulate wildfire propagation but instead complete a risk assessment of potential wildfire ignitions given the conditions of the terrain and the area's previous history of wildfires.

The WSGA estimates the location of ignition points and propagation durations of a wildfire given its burn scar. The WSGA could help fire investigators find what started a wildfire if debris is present at ignition points. This data could assist with determining the risk of future wildfires in that area.

7.3.4 Environmental Impact and Recovery

There are also other aspects of wildfire modelling that could be explored. One possible avenue would be the monitoring and simulation of environmental recovery of both fauna and flora; a time series of experimental data could be gathered using methods such as edge computing [113].

7.3.5 Closing Remark

Overall, the subject of wildfire modelling has an abundance of possible research areas with potential for innovation and improvement.

Appendix

This section includes additional content which may provide extra context to text from the main body of the thesis.

A.1 Generating a Land Cover Map using Satellite Data (Credit: Gourav Misra)

A nearly cloud-free satellite image (product) taken by the Sentinel-2 satellite in April 2022 was downloaded from the European Space Agency (ESA) Sentinel Hub portal, a satellite image hosting platform [205]. Sentinel-2 products contain multiple bands which capture electromagnetic radiation (light) reflected off Earth's surface. A Level-1C product (top of atmosphere reflectance values) was used in this study. The data were then atmospherically corrected to bottom-of-atmosphere values using the Semi-Automatic Classification Plugin in the geographic information software, QGIS which follows the dark-object subtraction (DOS) algorithm [206] [207] [208]. This is due to Level-1C products originally containing information regarding both Earth's surface and atmosphere which may not accurately depict the actual conditions on the surface of Earth. Three additional indices for highlighting vegetation, urban areas, and soil (i.e. normalised difference vegetation index (NDVI), normalised difference built-up index (NDBI), and normalised difference tillage index (NDTI) respectively) were generated from the original Sentinel-2 bands. These indices are simple ratios calculated from the original multispectral bands and are known to help distinguish various land cover classes, i.e. NDVI for vegetation, NDBI for urban areas, and NDTI for bare soil. NDVI is a ratio between the red and near-infrared bands whereas NDBI is calculated from the shortwave infrared (SWIR2) and near-infrared bands [209] [210]. The tillage index (NDTI) is calculated from both the available SWIR bands (SWIR1 and SWIR2) [209]. Areas of known land cover in the Wicklow Mountains, Ireland, were digitised to generate 70 polygons in total, which were further equally split into two sets. The first set was used to train a model, and the other was used as an independent validation of the predicted land cover class map. The training and validation datasets had 35 polygons each, which totalled to an area of 1.45km^2 and 1.4km^2 respectively. The polygons were selected

in a manner to ensure a coverage of the entire study area, with separation between the polygons (avoiding extremely close areas for both training and validation) and intra-class variability for avoiding bias in the model. A supervised random forest algorithm was applied on the stack of seven original bands (red, green, blue, near infrared (NIR), NIR narrow, short-wave infrared (SWIR1 and SWIR2)) captured by the Sentinel-2 satellite and the three additional indices to predict the land cover classes. The random forest classification algorithm is non-parametric in nature and known to improve estimates by averaging outputs from multiple decision trees to randomly subset from the data. It has been successfully used for land cover mapping in previous studies, and the algorithm was run using the RStoolbox library in the R programming environment [211] [212] [213]. The cloud-covered pixels in the area were classified based on a simple thresholding of the blue band [214]. The predicted land cover map was compared against the validation dataset which provided an overall accuracy of 97%. The predicted land cover map for the area under study is shown in Figure 3.1.

A.2 Figures for all WSGA Results

For each of the IGS generated burn scars described in Table 4.2, a comparison of the polygonal input burn scar and the WSGA seeded burn scars were recorded along with the fitness score of the best chromosome in the WSGA for each generation in Figure A.1, Figure A.2, Figure A.3, Figure A.4, Figure A.5, Figure A.6, Figure A.7, Figure A.8, Figure A.9, and Figure A.10.

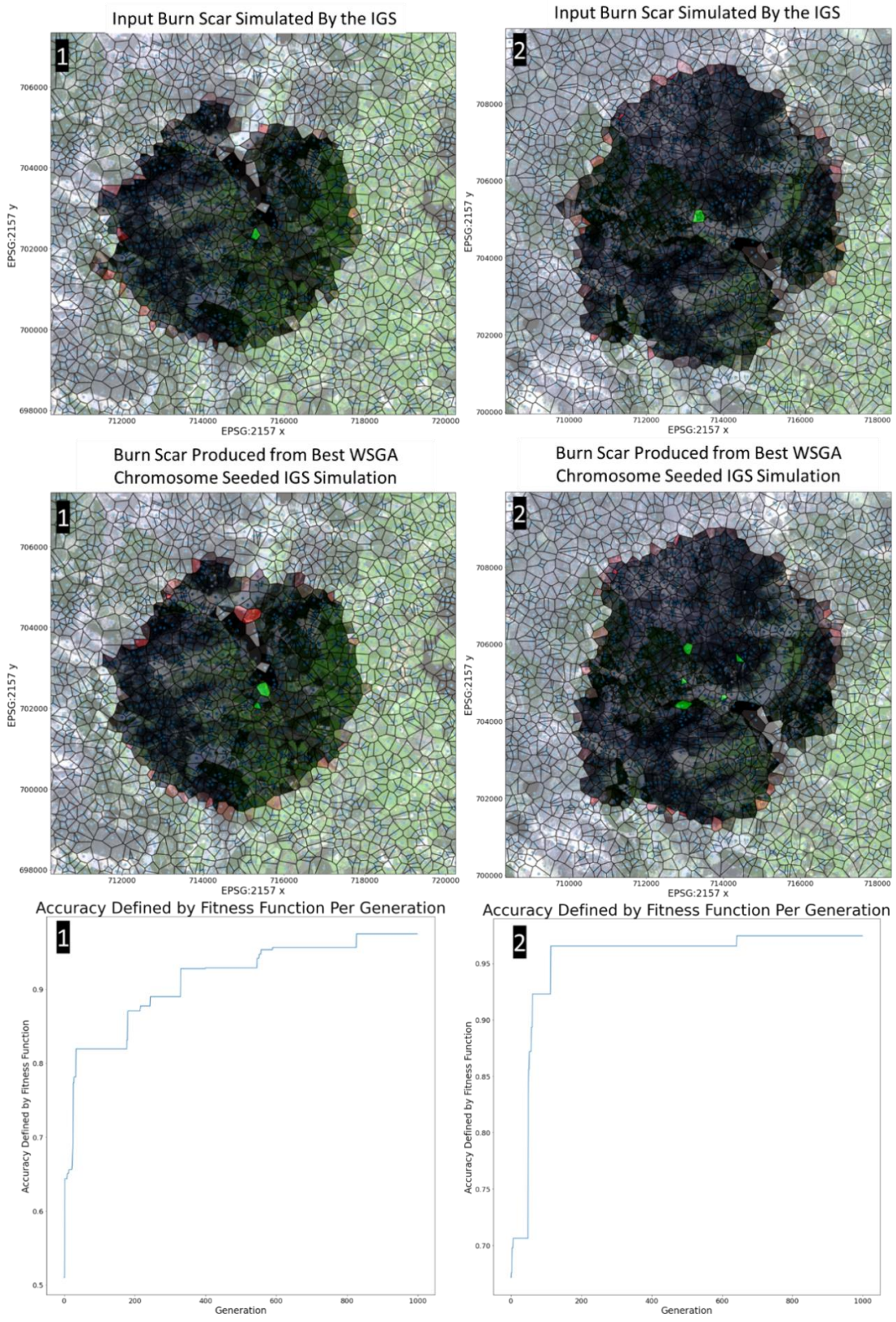

Figure A.1: An example of the results obtained using the WSGA. The top row shows the input burn scars of Wildfire 1 and Wildfire 2 respectively, simulated using the IGS. The middle row shows the burn scars created by simulating the wildfires from the best chromosome produced by the WSGA using the IGS. The condition of the polygons is represented by their colour, not burnt (transparent), wildfire source (green), burning (red) and burnt (black) based on the mass of the polygon in that state, with a satellite image of the terrain in the background (Copernicus 2022). The visual outputs are cropped to show source polygons, but the simulations may continue using the grid outside of this cropped zone. The bottom row shows the fitness score of the best chromosome at every generation of the WSGA for these wildfires.

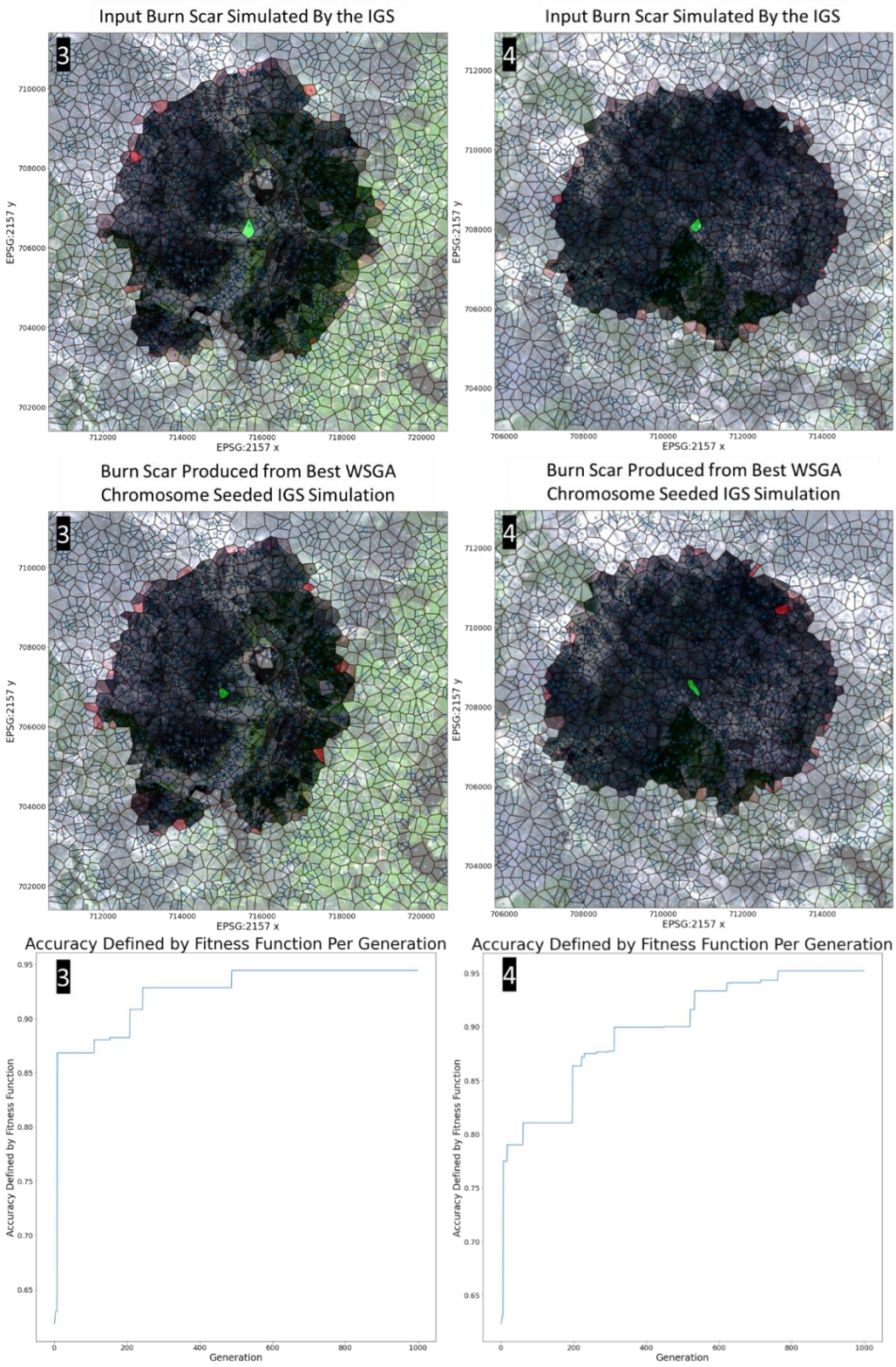

Figure A.2: An example of the results obtained using the WSGA. The top row shows the input burn scars of Wildfire 3 and Wildfire 4 respectively, simulated using the IGS. The middle row shows the burn scars created by simulating the wildfires from the best chromosome produced by the WSGA using the IGS. The condition of the polygons is represented by their colour, not burnt (transparent), wildfire source (green), burning (red) and burnt (black) based on the mass of the polygon in that state, with a satellite image of the terrain in the background (Copernicus 2022). The visual outputs are cropped to show source polygons, but the simulations may continue using the grid outside of this cropped zone. The bottom row shows the fitness score of the best chromosome at every generation of the WSGA for these wildfires.

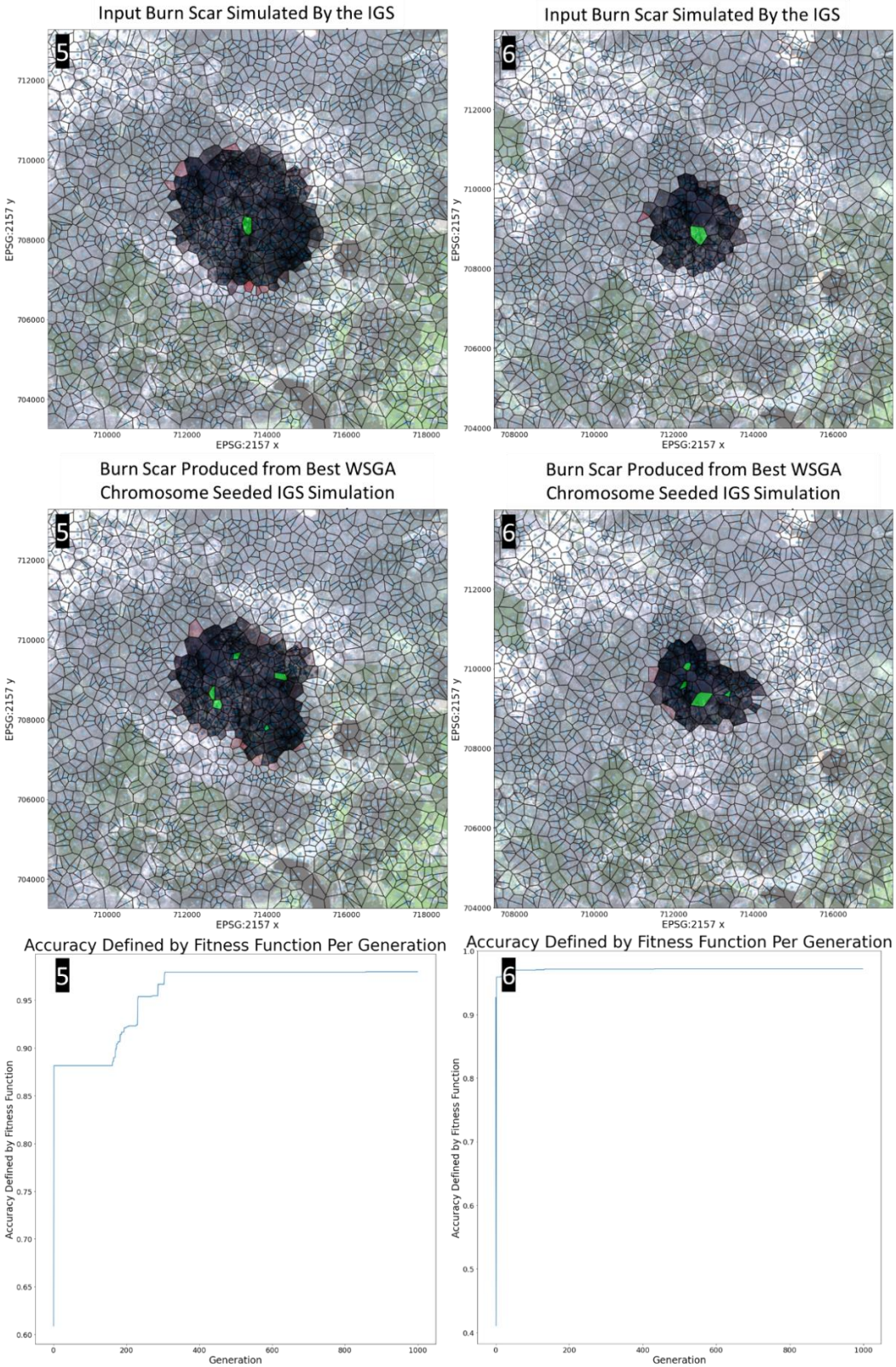

Figure A.3: An example of the results obtained using the WSGA. The top row shows the input burn scars of Wildfire 5 and Wildfire 6 respectively, simulated using the IGS. The middle row shows the burn scars created by simulating the wildfires from the best chromosome produced by the WSGA using the IGS. The condition of the polygons is represented by their colour, not burnt (transparent), wildfire source (green), burning (red) and burnt (black) based on the mass of the polygon in that state, with a satellite image of the terrain in the background (Copernicus 2022). The visual outputs are cropped to show source polygons, but the simulations may continue using the grid outside of this cropped zone. The bottom row shows the fitness score of the best chromosome at every generation of the WSGA for these wildfires.

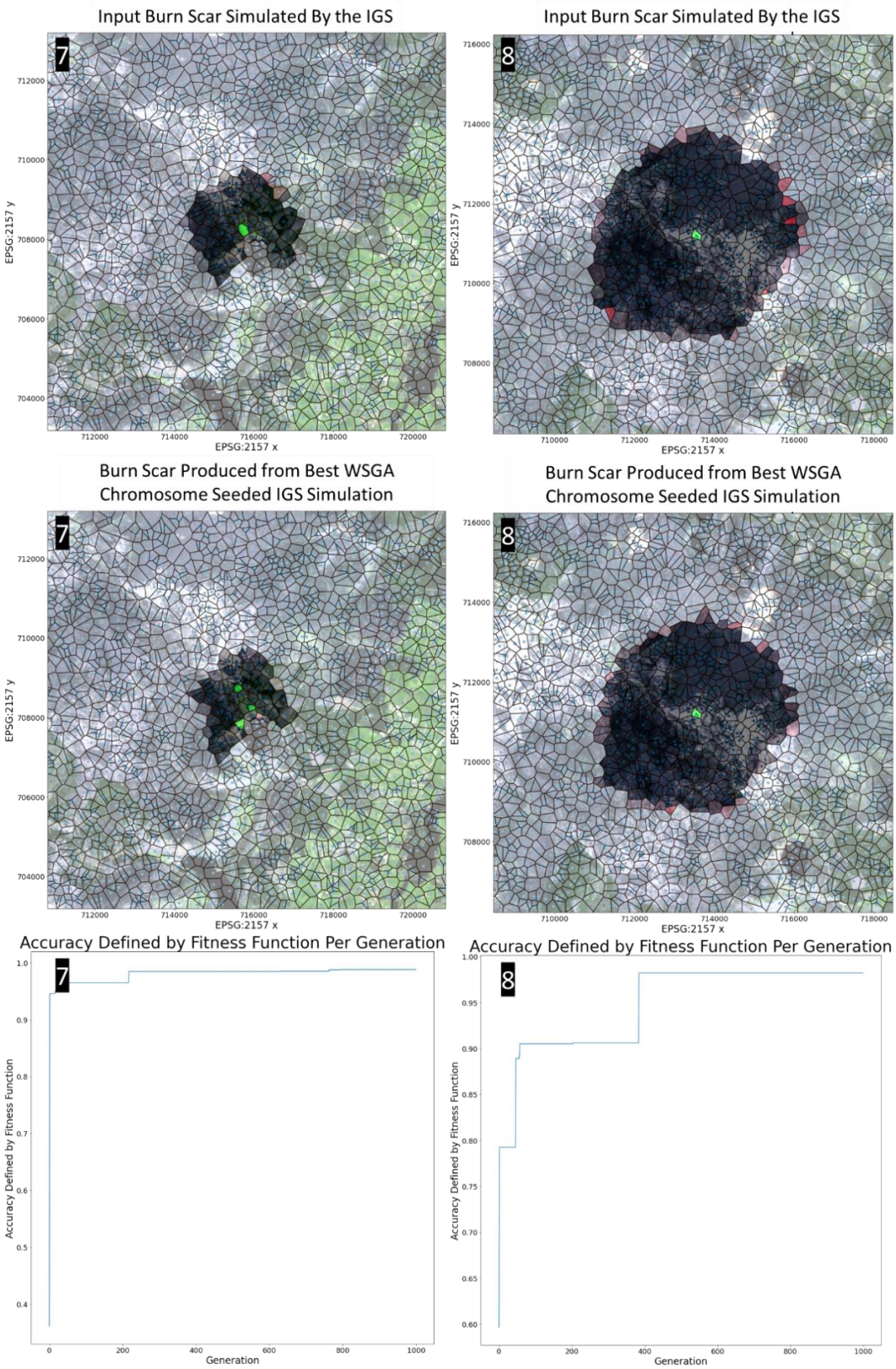

Figure A.4: An example of the results obtained using the WSGA. The top row shows the input burn scars of Wildfire 7 and Wildfire 8 respectively, simulated using the IGS. The middle row shows the burn scars created by simulating the wildfires from the best chromosome produced by the WSGA using the IGS. The condition of the polygons is represented by their colour, not burnt (transparent), wildfire source (green), burning (red) and burnt (black) based on the mass of the polygon in that state, with a satellite image of the terrain in the background (Copernicus 2022). The visual outputs are cropped to show source polygons, but the simulations may continue using the grid outside of this cropped zone. The bottom row shows the fitness score of the best chromosome at every generation of the WSGA for these wildfires.

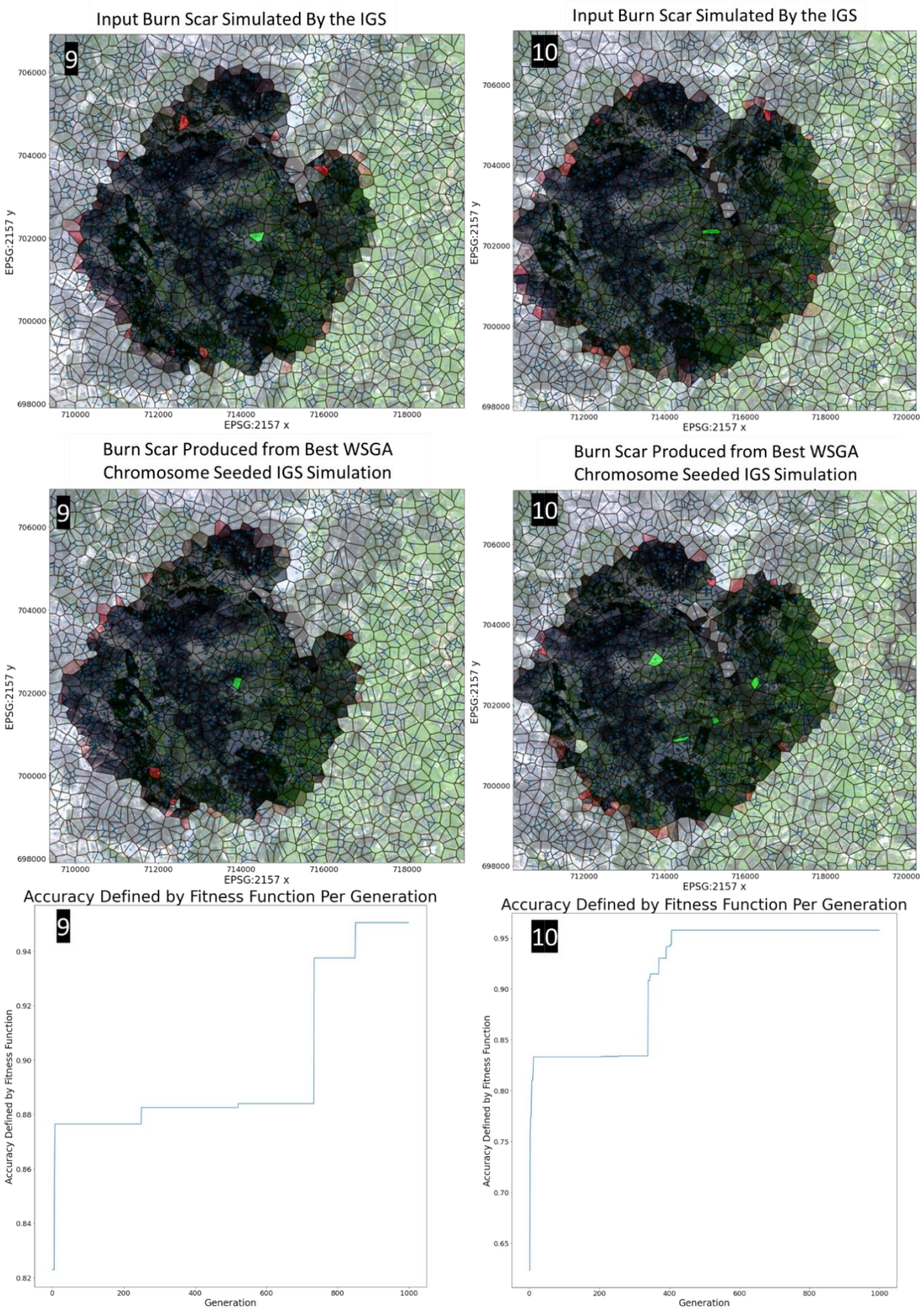

Figure A.5: An example of the results obtained using the WSGA. The top row shows the input burn scars of Wildfire 9 and Wildfire 10 respectively, simulated using the IGS. The middle row shows the burn scars created by simulating the wildfires from the best chromosome produced by the WSGA using the IGS. The condition of the polygons is represented by their colour, not burnt (transparent), wildfire source (green), burning (red) and burnt (black) based on the mass of the polygon in that state, with a satellite image of the terrain in the background (Copernicus 2022). The visual outputs are cropped to show source polygons, but the simulations may continue using the grid outside of this cropped zone. The bottom row shows the fitness score of the best chromosome at every generation of the WSGA for these wildfires.

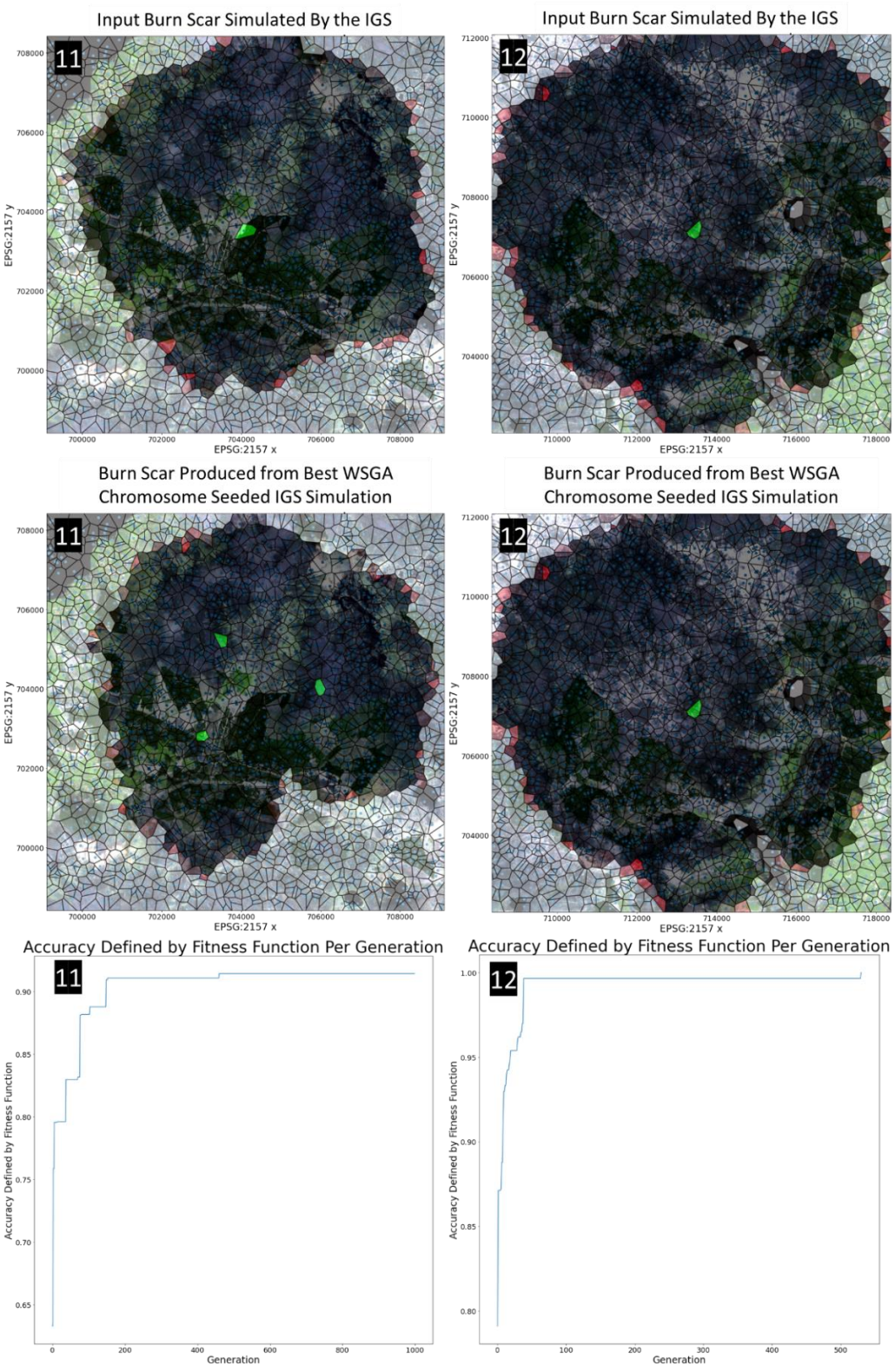

Figure A.6: An example of the results obtained using the WSGA. The top row shows the input burn scars of Wildfire 11 and Wildfire 12 respectively, simulated using the IGS. The middle row shows the burn scars created by simulating the wildfires from the best chromosome produced by the WSGA using the IGS. The condition of the polygons is represented by their colour, not burnt (transparent), wildfire source (green), burning (red) and burnt (black) based on the mass of the polygon in that state, with a satellite image of the terrain in the background (Copernicus 2022). The visual outputs are cropped to show source polygons, but the simulations may continue using the grid outside of this cropped zone. The bottom row shows the fitness score of the best chromosome at every generation of the WSGA for these wildfires.

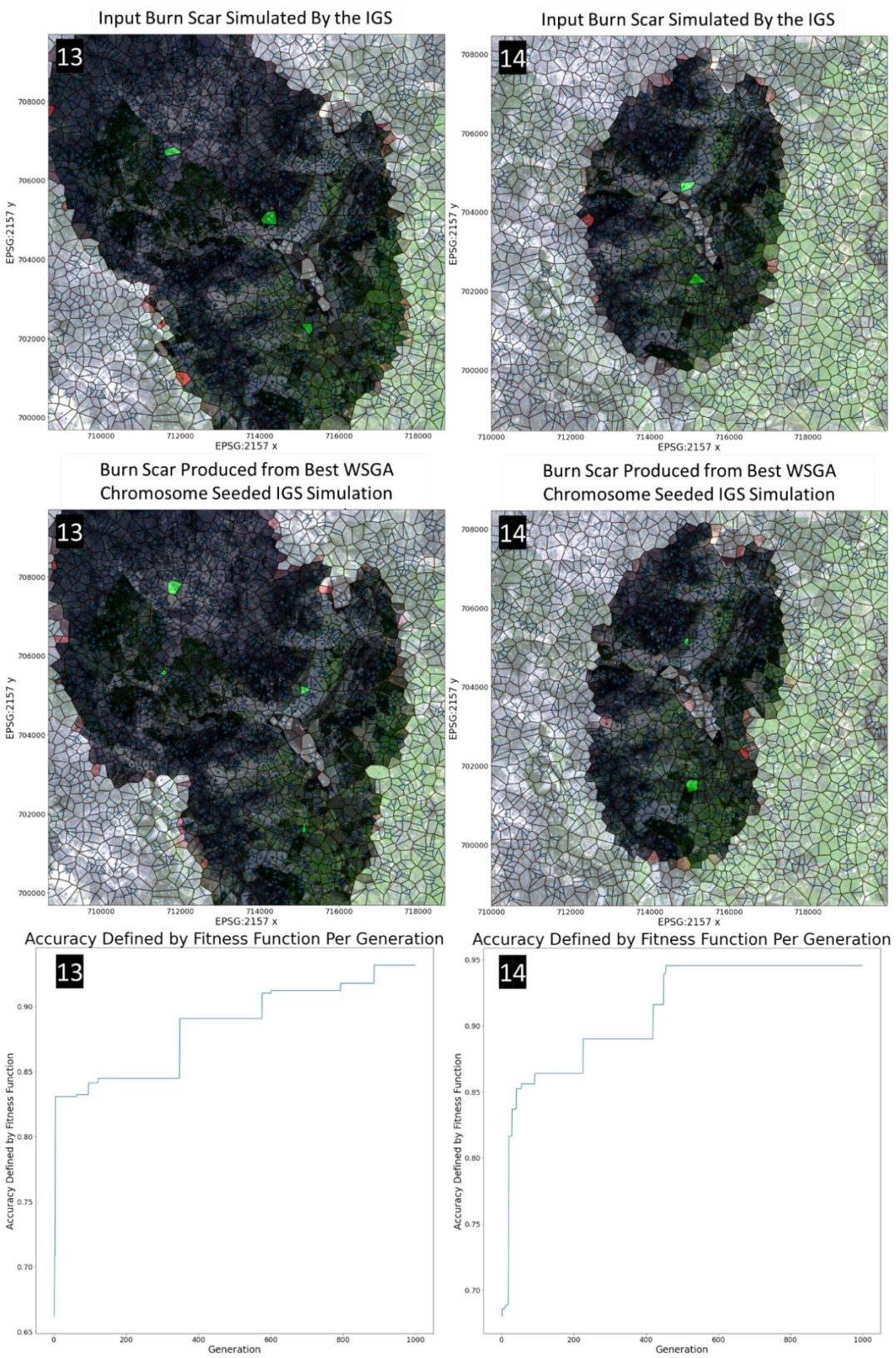

Figure A.7: An example of the results obtained using the WSGA. The top row shows the input burn scars of Wildfire 13 and Wildfire 14 respectively, simulated using the IGS. The middle row shows the burn scars created by simulating the wildfires from the best chromosome produced by the WSGA using the IGS. The condition of the polygons is represented by their colour, not burnt (transparent), wildfire source (green), burning (red) and burnt (black) based on the mass of the polygon in that state, with a satellite image of the terrain in the background (Copernicus 2022). The visual outputs are cropped to show source polygons, but the simulations may continue using the grid outside of this cropped zone. The bottom row shows the fitness score of the best chromosome at every generation of the WSGA for these wildfires.

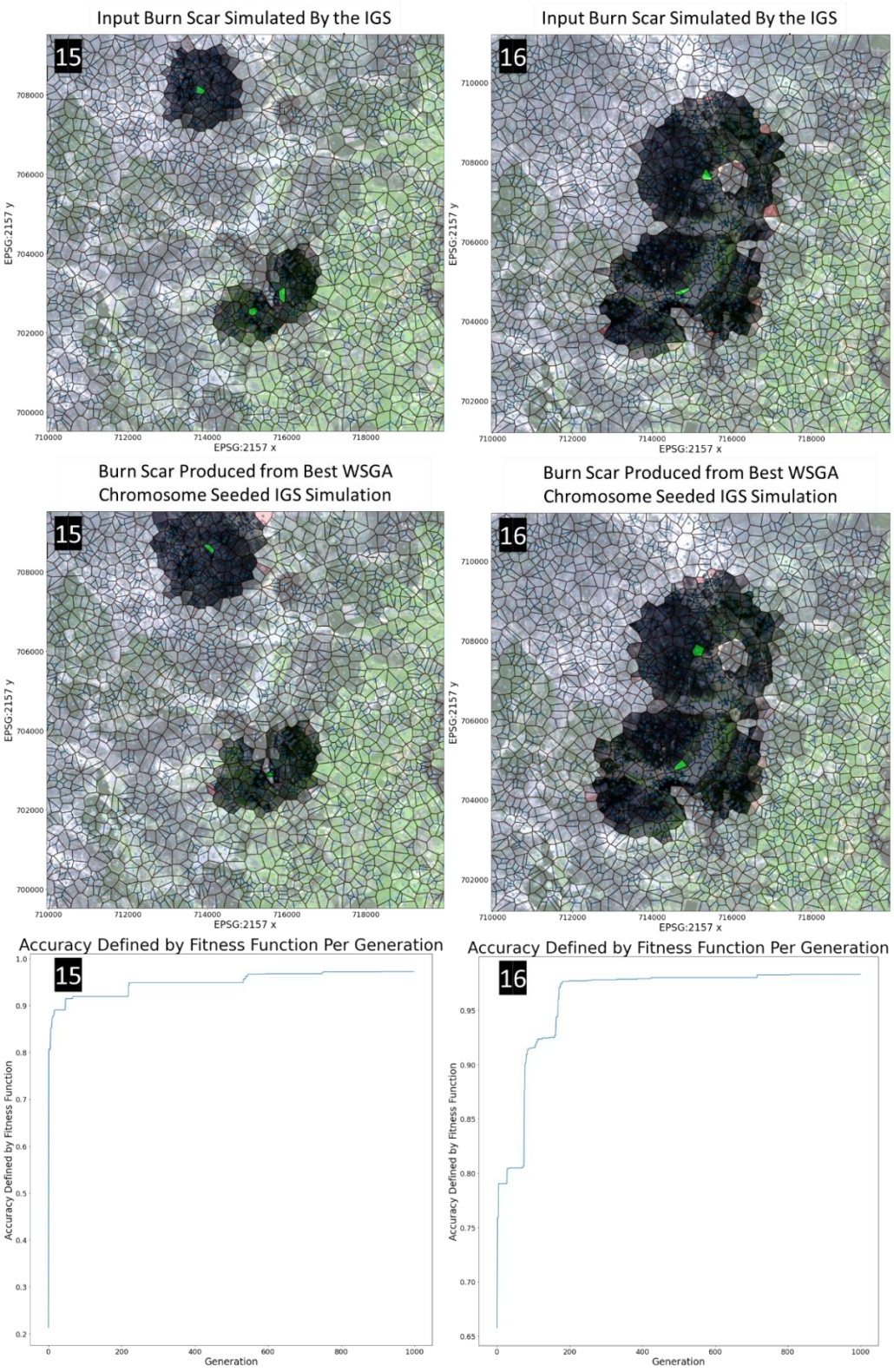

Figure A.8: An example of the results obtained using the WSGA. The top row shows the input burn scars of Wildfire 15 and Wildfire 16 respectively, simulated using the IGS. The middle row shows the burn scars created by simulating the wildfires from the best chromosome produced by the WSGA using the IGS. The condition of the polygons is represented by their colour, not burnt (transparent), wildfire source (green), burning (red) and burnt (black) based on the mass of the polygon in that state, with a satellite image of the terrain in the background (Copernicus 2022). The visual outputs are cropped to show source polygons, but the simulations may continue using the grid outside of this cropped zone. The bottom row shows the fitness score of the best chromosome at every generation of the WSGA for these wildfires.

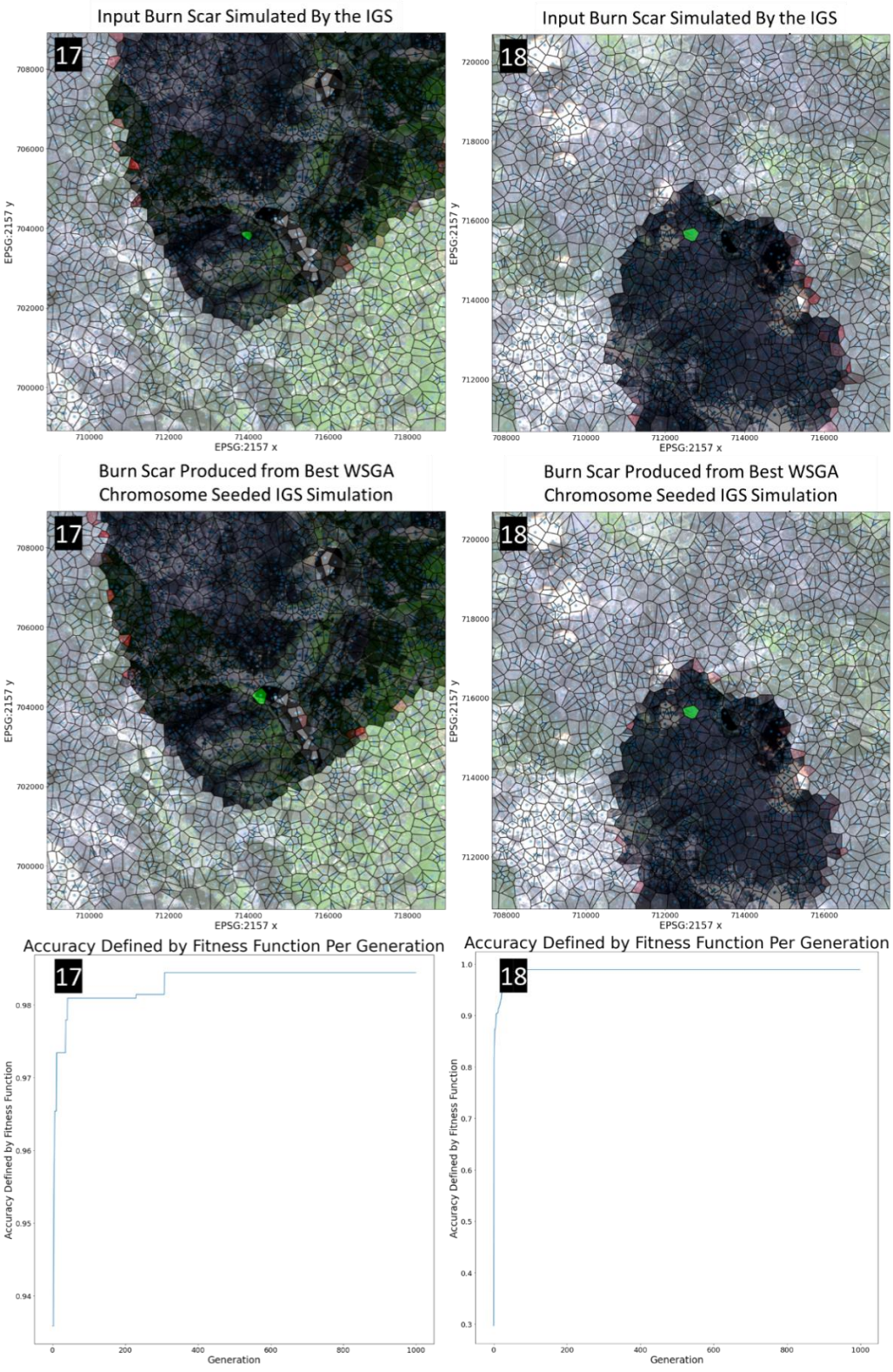

Figure A.9: An example of the results obtained using the WSGA. The top row shows the input burn scars of Wildfire 17 and Wildfire 18 respectively, simulated using the IGS. The middle row shows the burn scars created by simulating the wildfires from the best chromosome produced by the WSGA using the IGS. The condition of the polygons is represented by their colour, not burnt (transparent), wildfire source (green), burning (red) and burnt (black) based on the mass of the polygon in that state, with a satellite image of the terrain in the background (Copernicus 2022). The visual outputs are cropped to show source polygons, but the simulations may continue using the grid outside of this cropped zone. The bottom row shows the fitness score of the best chromosome at every generation of the WSGA for these wildfires.

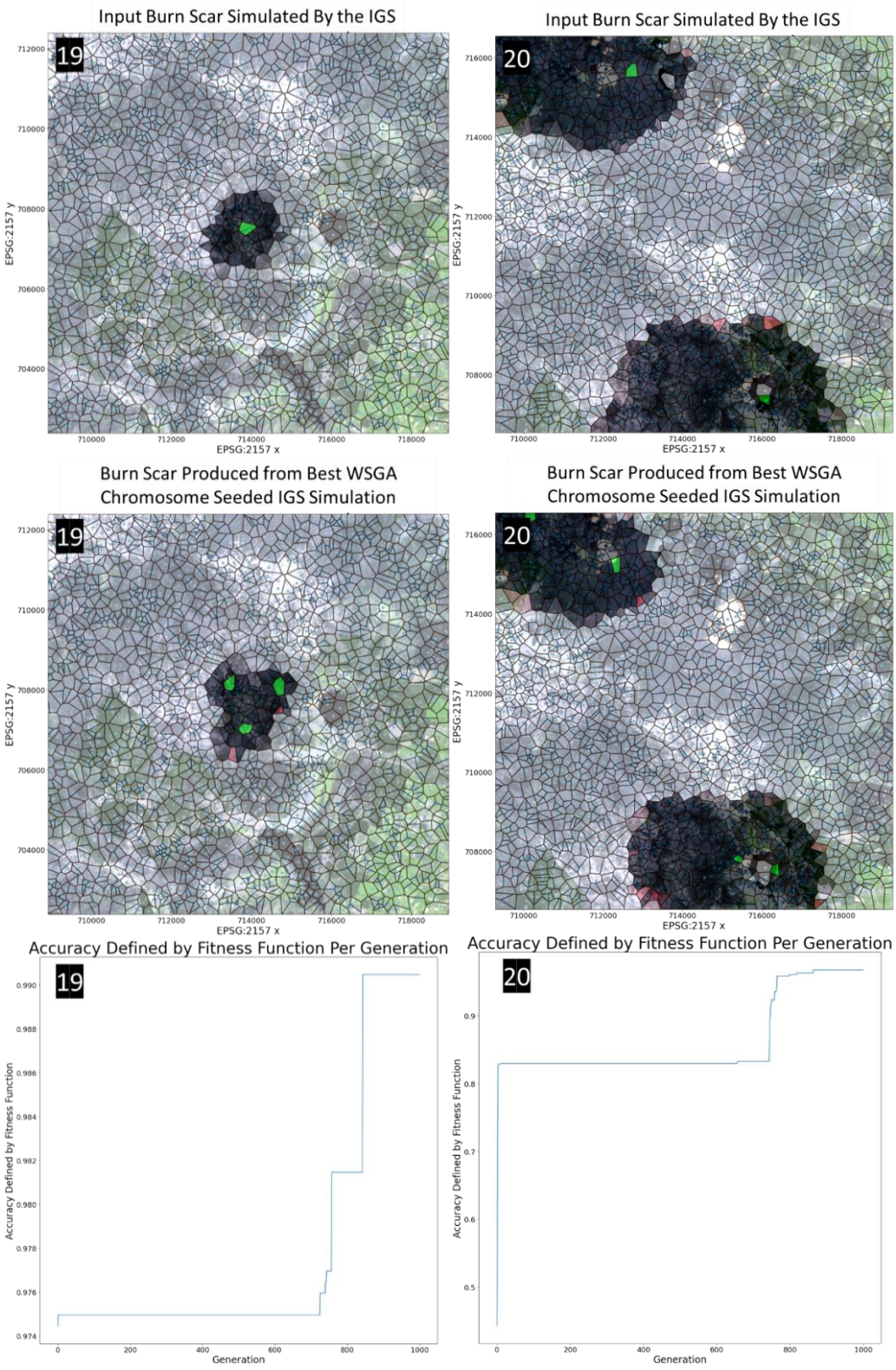

Figure A.10: An example of the results obtained using the WSGA. The top row shows the input burn scars of Wildfire 19 and Wildfire 20 respectively, simulated using the IGS. The middle row shows the burn scars created by simulating the wildfires from the best chromosome produced by the WSGA using the IGS. The condition of the polygons is represented by their colour, not burnt (transparent), wildfire source (green), burning (red) and burnt (black) based on the mass of the polygon in that state, with a satellite image of the terrain in the background (Copernicus 2022). The visual outputs are cropped to show source polygons, but the simulations may continue using the grid outside of this cropped zone. The bottom row shows the fitness score of the best chromosome at every generation of the WSGA for these wildfires.

A.3 Anemometer Validation

This section compares the two anemometers used in Chapter 5 (Table A.1).

Parkside Anemometer PWN A1 (Recorded Wind Speed in m/s)	RS PRO IM-740 Rotary Vane Anemometer (Recorded Wind Speed in m/s)
0	0.5
1	1.3
2	1.9
3	2.9
4.1	4.2
5.1	5.1
6	5.8

Table A.1: Comparison of the Parkside Anemometer PWN A1 and RS PRO IM-740 Rotary Vane Anemometer at different wind speeds.

A.4 Thesis Statistics

- Thesis Word Count: 59,659.
- Thesis Page Count: 223.
- Number of Chapters: 7.
- Number of Pages in Each Chapter: Chapter 1 has 6 pages, Chapter 2 has 34 pages, Chapter 3 has 32 pages, Chapter 4 has 33 pages, Chapter 5 has 36 pages, Chapter 6 has 13 pages and Chapter 7 has 7 pages.
- Areas Worked on During the Thesis: Wildfire Modelling, Frameworks for Modelling Propagation, Physical Fire Models, Fire and Rescue Services, Geographic Information Systems, Computer Vision, Genetic Algorithms, Experimental Apparatus Design, Chemistry of Combustion, and Data Analysis.
- Journal Publications: 1 published in Natural Hazards and Earth System Sciences, 1 accepted for publication in Fire Technology, and 1 submitted to Physics Education.
- Conference Publications: 1 published at the Irish Machine Vision and Image Processing Conference 2025.
- Book Chapters: 1 published in Modelling Insect Populations in Agricultural Landscapes.
- Number of Presentations Completed: 30.

Bibliography

[1] M. Haghani, E. Kuligowski, A. Rajabifard, and C. A. Kolden, “The state of wildfire and bushfire science: Temporal trends, research divisions and knowledge gaps,” *Saf Sci*, vol. 153, p. 105797, Sep. 2022, doi: 10.1016/j.ssci.2022.105797.

[2] H. Park, K. Nam, and H. Lim, “Is critical infrastructure safe from wildfires? A case study of wildland-industrial and -urban interface areas in South Korea,” *International Journal of Disaster Risk Reduction*, vol. 95, p. 103849, Sep. 2023, doi: 10.1016/j.ijdrr.2023.103849.

[3] F. Sari, “Identifying anthropogenic and natural causes of wildfires by maximum entropy method-based ignition susceptibility distribution models,” *J For Res (Harbin)*, vol. 34, no. 2, pp. 355–371, Apr. 2023, doi: 10.1007/s11676-022-01502-4.

[4] M. dos Reis, P. M. L. de A. Graça, A. M. Yanai, C. J. P. Ramos, and P. M. Fearnside, “Forest fires and deforestation in the central Amazon: Effects of landscape and climate on spatial and temporal dynamics,” *J Environ Manage*, vol. 288, p. 112310, Jun. 2021, doi: 10.1016/j.jenvman.2021.112310.

[5] J. T. Abatzoglou, A. P. Williams, L. Boschetti, M. Zubkova, and C. A. Kolden, “Global patterns of interannual climate–fire relationships,” *Glob Chang Biol*, vol. 24, no. 11, pp. 5164–5175, Nov. 2018, doi: 10.1111/gcb.14405.

[6] J. E. Halofsky, D. L. Peterson, and B. J. Harvey, “Changing wildfire, changing forests: the effects of climate change on fire regimes and vegetation in the Pacific Northwest, USA,” *Fire Ecology*, vol. 16, no. 1, p. 4, Dec. 2020, doi: 10.1186/s42408-019-0062-8.

[7] J. Piñol, K. Beven, and D. X. Viegas, “Modelling the effect of fire-exclusion and prescribed fire on wildfire size in Mediterranean ecosystems,” *Ecol Modell*, vol. 183, no. 4, pp. 397–409, May 2005, doi: 10.1016/j.ecolmodel.2004.09.001.

[8] United Nations Environment Programme & GRID-Arendal, “Spreading like Wildfire: The Rising Threat of Extraordinary Landscape Fires https://wedocs.unep.org/20.500.11822/38372,” 2022. Accessed: Mar. 13, 2024. [Online]. Available: https://wedocs.unep.org/20.500.11822/38372

[9] D. Dissing and D. L. Verbyla, “Spatial patterns of lightning strikes in interior Alaska and their relations to elevation and vegetation,” *Canadian Journal of Forest Research*, vol. 33, no. 5, pp. 770–782, May 2003, doi: 10.1139/x02-214.

[10] B. C. Bates, L. McCaw, and A. J. Dowdy, “Exploratory analysis of lightning-ignited wildfires in the Warren Region, Western Australia,” *J Environ Manage*, vol. 225, pp. 336–345, Nov. 2018, doi: 10.1016/j.jenvman.2018.07.097.

[11] H. Moradi, S. Fattorini, and J. Oldeland, “Influence of elevation on the species–area relationship,” *J Biogeogr*, vol. 47, no. 9, pp. 2029–2041, Sep. 2020, doi: 10.1111/jbi.13851.

[12] G. B. Bonan, H. H. Shugart, and D. L. Urban, “The sensitivity of some high-latitude boreal forests to climatic parameters,” *Clim Change*, vol. 16, no. 1, pp. 9–29, Feb. 1990, doi: 10.1007/BF00137344.

[13] Z. Liu, J. Yang, Y. Chang, P. J. Weisberg, and H. S. He, “Spatial patterns and drivers of fire occurrence and its future trend under climate change in a boreal forest of Northeast China,” *Glob Chang Biol*, vol. 18, no. 6, pp. 2041–2056, Jun. 2012, doi: 10.1111/j.1365-2486.2012.02649.x.

[14] W. Smith, T. Neale, and J. K. Weir, “Persuasion without policies: The work of reviving Indigenous peoples’ fire management in southern Australia,” *Geoforum*, vol. 120, pp. 82–92, Mar. 2021, doi: 10.1016/j.geoforum.2021.01.015.

[15] Y. Zhang *et al.*, “Impact of underground wildfire on plant roots in peat soil,” *Catena (Amst)*, vol. 259, p. 109371, Nov. 2025, doi: 10.1016/j.catena.2025.109371.

[16] R. T. Graham, S. McCaffrey, and T. B. Jain, “Science basis for changing forest structure to modify wildfire behavior and severity. . Gen. Tech. Rep. RMRS-GTR-120. Fort Collins, CO: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station. 43 p.,” 2004. doi: 10.2737/RMRS-GTR-120.

[17] A. Mendez and M. Farazmand, “Quantifying rare events in spotting: How far do wildfires spread?,” *Fire Saf J*, vol. 132, p. 103630, Sep. 2022, doi: 10.1016/j.firesaf.2022.103630.

[18] R. Blanchi, J. Leonard, K. Haynes, K. Opie, M. James, and F. D. de Oliveira, “Environmental circumstances surrounding bushfire fatalities in Australia 1901–2011,” *Environ Sci Policy*, vol. 37, pp. 192–203, Mar. 2014, doi: 10.1016/j.envsci.2013.09.013.

[19] A. Filkov, T. Duff, and T. Penman, “Improving Fire Behaviour Data Obtained from Wildfires,” *Forests*, vol. 9, no. 2, p. 81, Feb. 2018, doi: 10.3390/f9020081.

[20] E. Grant and J. D. Runkle, “Long-term health effects of wildfire exposure: A scoping review,” *The Journal of Climate Change and Health*, vol. 6, p. 100110, May 2022, doi: 10.1016/j.joclim.2021.100110.

[21] R. Skinner *et al.*, “A Literature Review on the Impact of Wildfires on Emergency Departments: Enhancing Disaster Preparedness,” *Prehosp Disaster Med*, vol. 37, no. 5, pp. 657–664, Oct. 2022, doi: 10.1017/S1049023X22001054.

[22] C. P. Kala, “Environmental and socioeconomic impacts of forest fires: A call for multilateral cooperation and management interventions,” *Natural Hazards Research*, vol. 3, no. 2, pp. 286–294, Jun. 2023, doi: 10.1016/j.nhres.2023.04.003.

[23] K. Gajendiran, S. Kandasamy, and M. Narayanan, “Influences of wildfire on the forest ecosystem and climate change: A comprehensive study,” *Environ Res*, vol. 240, p. 117537, Jan. 2024, doi: 10.1016/j.envres.2023.117537.

[24] C. Xue *et al.*, “A study on wildfire impacts on greenhouse gas emissions and regional air quality in South of Orléans, France,” *Journal of Environmental Sciences*, vol. 135, pp. 521–533, Jan. 2024, doi: 10.1016/j.jes.2022.08.032.

[25] S. Meier, R. J. R. Elliott, and E. Strobl, “The regional economic impact of wildfires: Evidence from Southern Europe,” *J Environ Econ Manage*, vol. 118, p. 102787, Mar. 2023, doi: 10.1016/j.jeem.2023.102787.

[26] E. Conlisk, V. Butsic, A. D. Syphard, S. Evans, and M. Jennings, “Evidence of increasing wildfire damage with decreasing property price in Southern California fires,” *PLoS One*, vol. 19, no. 4, p. e0300346, Apr. 2024, doi: 10.1371/journal.pone.0300346.

[27] P. Pandey *et al.*, “A global outlook on increasing wildfire risk: Current policy situation and future pathways,” *Trees, Forests and People*, vol. 14, p. 100431, Dec. 2023, doi: 10.1016/j.tfp.2023.100431.

[28] N. Boegelsack, J. Withey, G. O’Sullivan, and D. McMartin, “A Critical Examination of the Relationship between Wildfires and Climate Change with Consideration of the Human Impact,” *J Environ Prot (Irvine, Calif)*, vol. 09, no. 05, pp. 461–467, 2018, doi: 10.4236/jep.2018.95028.

[29] R. G. Benyon, A. Inbar, G. J. Sheridan, and P. N. J. Lane, “Critical climate thresholds for fire in wet, temperate forests,” *For Ecol Manage*, vol. 537, p. 120911, Jun. 2023, doi: 10.1016/j.foreco.2023.120911.

[30] N. Prat-Guitart *et al.*, “Peat Fires in Ireland,” in *Coal and Peat Fires: A Global Perspective*, Elsevier, 2019, pp. 451–482. doi: 10.1016/B978-0-12-849885-9.00020-2.

[31] P. Foss and P. Crushell, “Monaghan Fen Survey 2007,” *Internal Report for the Monaghan County Council & National Parks and Wildlife Service, Department of the Environment, Heritage and Local Government, Ireland*, 2007.

[32] I. Urák, N. Gallé-Szpisjak, I.-L. Szigyártó, A.-R. Zsigmond, and R. Gallé, “Heather (Calluna vulgaris) supports spider diversity of oligotrophic peat bogs,” *J Insect Conserv*, vol. 27, no. 3, pp. 415–422, Jun. 2023, doi: 10.1007/s10841-023-00465-z.

[33] G. M. Davies and C. J. Legg, “Developing a live fuel moisture model for moorland fire danger rating,” Aug. 2008, pp. 225–236. doi: 10.2495/FIVA080231.

[34] E. J. Fusco, J. T. Finn, J. K. Balch, R. C. Nagy, and B. A. Bradley, “Invasive grasses increase fire occurrence and frequency across US ecoregions,” *Proceedings of the National Academy of Sciences*, vol. 116, no. 47, pp. 23594–23599, Nov. 2019, doi: 10.1073/pnas.1908253116.

[35] J. Madrigal *et al.*, “Do cone age and heating mode determine the opening of serotinous cones during wildfires? A new bench scale approach applied to Pinus halepensis Mill.,” *Science of The Total Environment*, vol. 763, p. 144222, Apr. 2021, doi: 10.1016/j.scitotenv.2020.144222.

[36] Copernicus, “Emergency Management Service.” Accessed: Jun. 11, 2025. [Online]. Available: https://forest-fire.emergency.copernicus.eu/apps/effis_current_situation/

[37] R. O. Weber, “Modelling fire spread through fuel beds,” *Prog Energy Combust Sci*, vol. 17, no. 1, pp. 67–82, Jan. 1991, doi: 10.1016/0360-1285(91)90003-6.

[38] A. G. McArthur, “Weather and grassland fire behaviour,” *1923-1978 & Australia. Forestry and Timber Bureau*, 1966.

[39] I. R. Noble, A. M. Gill, and G. A. V. Bary, “McArthur’s fire-danger meters expressed as equations,” *Australian Journal of Ecology*, vol. 5, no. 2, pp. 201–203, Jun. 1980, doi: 10.1111/j.1442-9993.1980.tb01243.x.

[40] A. Cardíl *et al.*, “Characterizing the rate of spread of large wildfires in emerging fire environments of northwestern Europe using Visible Infrared Imaging Radiometer Suite active fire data,” *Natural Hazards and Earth System Sciences*, vol. 23, no. 1, pp. 361–373, Feb. 2023, doi: 10.5194/nhess-23-361-2023.

[41] W. H. Frandsen, “Fire spread through porous fuels from the conservation of energy,” *Combust Flame*, vol. 16, no. 1, pp. 9–16, Feb. 1971, doi: 10.1016/S0010-2180(71)80005-6.

[42] W. L. Fons, “Analysis of Fire Spread in Light Forest Fuels,” *J Agric Res*, vol. 72, no. 3, pp. 92–121, 1946.

[43] DecaMap, “DecaMap.” Accessed: Feb. 22, 2024. [Online]. Available: https://decamap.com/

[44] K. G. Hirsch, “Canadian Forest Fire Behavior Prediction (FBP) System: user’s guide,” *Canadian Forest Service Northwest Region Northern Forestry Centre*, 1996.

[45] P. Helene, C. Britez, and M. Carvalho, “Fire impacts on concrete structures. A brief review,” *Revista ALCONPAT*, vol. 10, no. 1, pp. 1–21, Dec. 2019, doi: 10.21041/ra.v10i1.421.

[46] R. C. Rothermel, “A mathematical model for predicting fire spread in wildland fuels,” *Res. Pap. INT-115. Ogden, UT: U.S. Department of Agriculture, Intermountain Forest and Range Experiment Station. 40 p.*, 1972.

[47] P. L. Andrews, “The Rothermel surface fire spread model and associated developments: A comprehensive explanation,” *The Rothermel surface fire spread model and associated developments: A comprehensive explanation. Gen. Tech. Rep. RMRS-GTR-371. Fort Collins, CO: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station. 121 p.*, 2018.

[48] R. Wilson, “Reformulation of forest fire spread equations in SI units,” *Research Note INT-292. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Range and Forest Experiment Station. 5 p.*, 1980, doi: 10.2737/INT-RN-292.

[49] ForeFire, “ForeFire GitHub Page.” Accessed: Jun. 24, 2025. [Online]. Available: https://github.com/forefireAPI/forefire/tree/v1.1.10

[50] J. Ning *et al.*, “Comparison of Different Models to Simulate Forest Fire Spread: A Case Study,” *Forests*, vol. 15, no. 3, p. 563, Mar. 2024, doi: 10.3390/f15030563.

[51] R. R. Linn *et al.*, “QUIC-fire: A fast-running simulation tool for prescribed fire planning,” *Environmental Modelling & Software*, vol. 125, p. 104616, Mar. 2020, doi: 10.1016/j.envsoft.2019.104616.

[52] Z. Lin, H. H. T. Liu, and M. Wotton, “Kalman Filter-Based Large-Scale Wildfire Monitoring With a System of UAVs,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 1, pp. 606–615, Jan. 2019, doi: 10.1109/TIE.2018.2823658.

[53] S. L. Lauritzen, “Time Series Analysis in 1880: A Discussion of Contributions Made by T.N. Thiele,” *Int Stat Rev*, vol. 49, no. 3, p. 319, Dec. 1981, doi: 10.2307/1402616.

[54] M. E. Green, K. Kaiser, and N. Shenton, “Modeling Wildfire Perimeter Evolution using Deep Neural Networks,” Sep. 2020.

[55] M. Gardner, “Mathematical Games,” *Sci Am*, vol. 223, no. 4, pp. 120–123, Oct. 1970, doi: 10.1038/scientificamerican1070-120.

[56] C. Pais, J. Carrasco, D. L. Martell, A. Weintraub, and D. L. Woodruff, “Cell2Fire: A Cell-Based Forest Fire Growth Model to Support Strategic Landscape Management Planning,” *Frontiers in Forests and Global Change*, vol. 4, Nov. 2021, doi: 10.3389/ffgc.2021.692706.

[57] K. Małecki, “Graph Cellular Automata with Relation-Based Neighbourhoods of Cells for Complex Systems Modelling: A Case of Traffic Simulation,” *Symmetry (Basel)*, vol. 9, no. 12, p. 322, Dec. 2017, doi: 10.3390/sym9120322.

[58] S. Zhang, J. Liu, H. Gao, X. Chen, X. Li, and J. Hua, “Study on Forest Fire spread Model of Multi-dimensional Cellular Automata based on Rothermel Speed Formula,” *CERNE*, vol. 27, 2021, doi: 10.1590/01047760202127012932.

[59] M. A. Finney, “Modeling the spread and behavior of prescribed natural fires.,” in *Proceedings of the 12th International Conference on Fire and Forest Meteorology. 1993 October 26-28; Jekyll Island, GA.*, 1994.

[60] M. A. Finney, “FARSITE: Fire Area Simulator-model development and evaluation,” 1998. doi: 10.2737/RMRS-RP-4.

[61] M. E. Alexander, “Estimating the length-to-breadth ratio of elliptical forest fire patterns,” in *Proceedings of the 8th Conference on Fire and Forest Meteorology, April 29-May 2, 1985, Detroit, MI. Bethesda, MD: Society of American Foresters*, 1985, pp. 287–304.

[62] G. D. Richards, “An elliptical growth model of forest fire fronts and its numerical solution,” *Int J Numer Methods Eng*, vol. 30, no. 6, pp. 1163–1179, Oct. 1990, doi: 10.1002/nme.1620300606.

[63] C. E. Van Wagner, “Conditions for the start and spread of crown fire,” *Canadian Journal of Forest Research*, vol. 7, no. 1, pp. 23–34, Mar. 1977, doi: 10.1139/x77-004.

[64] C. E. Van Wagner, “Prediction of crown fire behavior in two stands of jack pine,” *Canadian Journal of Forest Research*, vol. 23, no. 3, pp. 442–449, Mar. 1993, doi: 10.1139/x93-062.

[65] C. Tymstra, R. W. Bryce, B. M. Wotton, S. W. Taylor, and O. B. Armitage, “Development and structure of Prometheus : the Canadian Wildland Fire Growth Simulation Model,” *Canadian Forest Service Northern Forestry Centre*, 2010.

[66] J.-B. Filippi, F. Bosseur, and D. Grandi, “ForeFire: open-source code for wildland fire spread models,” in *Advances in forest fire research*, Imprensa da Universidade de Coimbra, 2014, pp. 275–282. doi: 10.14195/978-989-26-0884-6_29.

[67] R. E. Burgan and R. C. Rothermel, “BEHAVE: fire behavior prediction and fuel modeling system--FUEL subsystem,” 1984. doi: 10.2737/INT-GTR-167.

[68] W. H. Frandsen, “Fire spread through porous fuels from the conservation of energy,” *Combust Flame*, vol. 16, no. 1, pp. 9–16, Feb. 1971, doi: 10.1016/S0010-2180(71)80005-6.

[69] R. C. Rothermel and H. . Anderson, “Fire spread characteristics determined in the laboratory,” *Res. Pap. INT-RP-30. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 34 p.*, 1966.

[70] L. A. Wood, “The Use of Dew-Point Temperature in Humidity Calculations ,” *JOURNAL OF RESEARCH of the National Bureau of Standards - C. Engineering and Instrumentation*, vol. 74C, no. 3, pp. 117–122, 1970.

[71] J. R. Martínez-de Dios, J. C. André, J. C. Gonçalves, B. Ch. Arrue, A. Ollero, and D. X. Viegas, “Laboratory fire spread analysis using visual and infrared images,” *Int J Wildland Fire*, vol. 15, no. 2, p. 179, 2006, doi: 10.1071/WF05004.

[72] W. R. CATCHPOLE, E. A. CATCHPOLE, B. W. BUTLER, R. C. ROTHERMEL, G. A. MORRIS, and D. J. LATHAM, “Rate of Spread of Free-Burning Fires in Woody Fuels in a Wind Tunnel,” *Combustion Science and Technology*, vol. 131, no. 1–6, pp. 1–37, Jan. 1998, doi: 10.1080/00102209808935753.

[73] N. Prat-Guitart, R. M. Hadden, C. M. Belcher, G. Rein, and J. M. Yearsley, “Propagation probability and spread rates of self-sustained smouldering fires under controlled

moisture content and bulk density conditions," *Int J Wildland Fire*, vol. 25, no. 4, pp. 456–465, Apr. 2016, doi: 10.1071/WF15103.

[74] N. Prat-Guitart, G. Rein, R. M. Hadden, C. M. Belcher, and J. M. Yearsley, "Infrared Image Analysis as a Tool for Studying the Horizontal Smoldering Propagation of Laboratory Peat Fires," in *Coal and Peat Fires: a Global Perspective*, Elsevier, 2015, pp. 121–139. doi: 10.1016/B978-0-444-59510-2.00006-9.

[75] A. S. Campbell, "Some Burning Characteristics of Filter Paper," *Combustion Science and Technology*, vol. 3, no. 3, pp. 103–120, May 1971, doi: 10.1080/00102207108952278.

[76] J. Zhang, Y.-C. Zhang, P. Alstrøm, and M. T. Levinsen, "Modeling forest fire by a paper-burning experiment, a realization of the interface growth mechanism," *Physica A: Statistical Mechanics and its Applications*, vol. 189, no. 3–4, pp. 383–389, Nov. 1992, doi: 10.1016/0378-4371(92)90050-Z.

[77] X. J. Wang, J. R. J. Thompson, W. J. Braun, and D. G. Woolford, "Fitting a stochastic fire spread model to data," *Adv Stat Climatol Meteorol Oceanogr*, vol. 5, no. 1, pp. 57–66, Apr. 2019, doi: 10.5194/ascmo-5-57-2019.

[78] W. J. Braun, "Assessing a Stochastic Fire Spread Simulator," *Journal of Environmental Informatics*, vol. 22, no. 1, pp. 1–12, Sep. 2013, doi: 10.3808/jei.201300241.

[79] Citizens Information, "Fire services." Accessed: May 09, 2025. [Online]. Available: https://www.citizensinformation.ie/en/government-in-ireland/security-and-emergency-services/fire-services/

[80] Civil Defence Cosaint Shibhialta, "Civil Defence." Accessed: Dec. 28, 2025. [Online]. Available: https://www.civildefence.ie/

[81] Óglaigh na hÉireann Irish Defence Forces, "Air Corps", Accessed: May 09, 2025. [Online]. Available: https://www.military.ie/en/who-we-are/air-corps/

[82] Óglaigh na hÉireann Irish Defence Forces, "Air Corps Fleet." Accessed: May 09, 2025. [Online]. Available: https://www.military.ie/en/who-we-are/air-corps/the-fleet/

[83] SFI, "SFI-Defence Organisation Innovation Challenge, Challenge 1," https://web.archive.org/web/20221115145029/https://www.sfi.ie/funding/funding-calls/future-innovator-defence/#row-collapse-item-1. Accessed: Jan. 25, 2024. [Online]. Available:

https://web.archive.org/web/20221115145029/https://www.sfi.ie/funding/funding-calls/future-innovator-defence/#row-collapse-item-1

[84] Coillte, “Coillte.” Accessed: May 09, 2025. [Online]. Available: https://www.coillte.ie/

[85] S. A. Royle, “Beyond the boundaries in the island of Ireland,” *Journal of Marine and Island Cultures*, vol. 1, no. 2, pp. 91–98, Dec. 2012, doi: 10.1016/j.imic.2012.11.005.

[86] National Parks and Wildlife Service, “National Parks and Wildlife Service.” Accessed: May 09, 2025. [Online]. Available: https://www.npws.ie/

[87] National Parks and Wildlife Service, “National parks of Ireland.” Accessed: May 09, 2025. [Online]. Available: https://www.nationalparks.ie/

[88] M. Jurvélius, “HEALTH AND PROTECTION | Forest Fires (Prediction, Prevention, Preparedness and Suppression),” in *Encyclopedia of Forest Sciences*, Elsevier, 2004, pp. 334–339. doi: 10.1016/B0-12-145160-7/00277-5.

[89] A. A. G. Wilson, “Width of firebreak that is necessary to stop grass fires: some field experiments,” *Canadian Journal of Forest Research*, vol. 18, no. 6, pp. 682–687, Jun. 1988, doi: 10.1139/x88-104.

[90] M. García Hernández, “Coordination and Inclusion in the Surveillance, Prevention and Fighting of Wildfires in the South of the State of Puebla,” *Proceedings of the Fifth International Symposium on Fire Economics, Planning, and Policy: Ecosystem Services and Wildfires*, 2019.

[91] H. Hesseln, “Wildland Fire Prevention: a Review,” *Current Forestry Reports*, vol. 4, no. 4, pp. 178–190, Dec. 2018, doi: 10.1007/s40725-018-0083-6.

[92] A. D. Syphard and J. E. Keeley, “Location, timing and extent of wildfire vary by cause of ignition,” *Int J Wildland Fire*, vol. 24, no. 1, p. 37, 2015, doi: 10.1071/WF14024.

[93] J. K. Balch, B. A. Bradley, J. T. Abatzoglou, R. C. Nagy, E. J. Fusco, and A. L. Mahood, “Human-started wildfires expand the fire niche across the United States,” *Proceedings of the National Academy of Sciences*, vol. 114, no. 11, pp. 2946–2951, Mar. 2017, doi: 10.1073/pnas.1617394114.

[94] The National Wildfire Coordinating Group, *Guide to Wildland Fire Origin and Cause Determination (2016)*, 1st ed., vol. 1. 2019.

[95] A. B. Cox and F. Gifford, “An overview to geographic information systems,” *The Journal of Academic Librarianship*, vol. 23, no. 6, pp. 449–461, Nov. 1997, doi: 10.1016/S0099-1333(97)90169-5.

[96] epsg.io, “EPSG:2157.” Accessed: May 10, 2025. [Online]. Available: https://epsg.io/2157

[97] Copernicus, “Copernicus Data Space Ecosystem.” Accessed: Jan. 25, 2024. [Online]. Available: https://dataspace.copernicus.eu/

[98] European Space Agency, “Sentinel-2”, Accessed: May 10, 2025. [Online]. Available: https://www.esa.int/Applications/Observing_the_Earth/Copernicus/Sentinel-2

[99] ESA, “Create DEM-related Bands”, Accessed: May 10, 2025. [Online]. Available: https://step.esa.int/main/wp-content/help/versions/9.0.0/snap/org.esa.snap.snap.help/desktop/CreateDemRelatedBands.html

[100] ESA, “Download - STEP.” Accessed: Jan. 25, 2024. [Online]. Available: https://step.esa.int/main/download/

[101] CORINE, “Home :: CORINE Land Cover classes,” https://land.copernicus.eu/content/corine-land-cover-nomenclature-guidelines/html/. Accessed: Jan. 25, 2024. [Online]. Available: https://land.copernicus.eu/content/corine-land-cover-nomenclature-guidelines/html/

[102] Met Éireann, “Met Éireann”, Accessed: May 10, 2025. [Online]. Available: https://www.met.ie/

[103] Copernicus, “Data and services.” Accessed: Feb. 10, 2025. [Online]. Available: https://forest-fire.emergency.copernicus.eu/applications/data-and-services

[104] R. Szeliski, *Computer Vision*. Cham: Springer International Publishing, 2022. doi: 10.1007/978-3-030-34372-9.

[105] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. 2004.

[106] H. Zhang, “Machine Vision: A Comprehensive Analysis of Techniques, Applications, and Challenges,” *Highlights in Science, Engineering and Technology*, vol. 71, pp. 299–304, Nov. 2023, doi: 10.54097/hset.v71i.13050.

[107] R. Manish, A. Venkatesh, and S. Denis Ashok, “Machine Vision Based Image Processing Techniques for Surface Finish and Defect Inspection in a Grinding Process,” *Mater Today Proc*, vol. 5, no. 5, pp. 12792–12802, 2018, doi: 10.1016/j.matpr.2018.02.263.

[108] A. Rosenfeld and J. L. Pfaltz, “Sequential Operations in Digital Picture Processing,” *Journal of the ACM*, vol. 13, no. 4, pp. 471–494, Oct. 1966, doi: 10.1145/321356.321357.

[109] Intel ®, “OpenCV.” Accessed: Mar. 06, 2025. [Online]. Available: https://opencv.org/

[110] D. Kelly, J. McDonald, and C. Markham, “A person independent system for recognition of hand postures used in sign language,” *Pattern Recognit Lett*, vol. 31, no. 11, pp. 1359–1368, Aug. 2010, doi: 10.1016/j.patrec.2010.02.004.

[111] N. J. Uke and R. C. Thool, “Moving Vehicle Detection for Measuring Traffic Count Using OpenCV,” *Journal of Automation and Control Engineering*, pp. 349–352, 2013, doi: 10.12720/joace.1.4.349-352.

[112] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” Jun. 2015.

[113] G. R. Palma, C. P. Hackett, and C. Markham, “Machine Vision Applied to Entomology,” 2023, pp. 149–184. doi: 10.1007/978-3-031-43098-5_9.

[114] K. O’Shea and R. Nash, “An Introduction to Convolutional Neural Networks,” Nov. 2015.

[115] X. Geng, Y. Su, X. Cao, H. Li, and L. Liu, “YOLOFM: an improved fire and smoke object detection algorithm based on YOLOv5n,” *Sci Rep*, vol. 14, no. 1, p. 4543, Feb. 2024, doi: 10.1038/s41598-024-55232-0.

[116] X. Qi and J. Ebert, “A Computer Vision-Based Method for Fire Detection in Color Videos,” *International Journal of Imaging and Robotics*, vol. 2, no. 9, 2009.

[117] U. Hwang and J. Jeong, “Computer Vision Based Smoke Detection Method by Using Colour and Object Tracking,” in *Computer Science & Information Technology (CS & IT)*, Academy & Industry Research Collaboration Center (AIRCC), Feb. 2016, pp. 53–59. doi: 10.5121/csit.2016.60306.

[118] G. Zhai *et al.*, “From spark to suppression: An overview of wildfire monitoring, progression prediction, and extinguishing techniques,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 140, p. 104600, Jun. 2025, doi: 10.1016/j.jag.2025.104600.

[119] Kaixi Li and Yingyun Yang, “Fire detection algorithm based on CLG-TV optical flow model,” in *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, IEEE, Oct. 2016, pp. 1381–1385. doi: 10.1109/CompComm.2016.7924930.

[120] X. Chen and Z. Shen, “Theories and Numerics of Inverse Problems,” Institute for Mathematical Sciences Event Archive. Accessed: Nov. 30, 2025. [Online]. Available: https://imsarchives.nus.edu.sg/oldwww2/events/2018/theo/index.html#:~:text=It%20(%20An%20inverse%20problem%20)%20is,including%20mathematics%2C%20engineering%2C%20medicine%2C%20physics%2C%20and%20geophysics.

[121] M. R. Fellows, F. V. Fomin, D. Lokshtanov, F. Rosamond, S. Saurabh, and Y. Villanger, “Local search: Is brute-force avoidable?,” *J Comput Syst Sci*, vol. 78, no. 3, pp. 707–719, May 2012, doi: 10.1016/j.jcss.2011.10.003.

[122] M. Sambridge and K. Mosegaard, “MONTE CARLO METHODS IN GEOPHYSICAL INVERSE PROBLEMS,” *Reviews of Geophysics*, vol. 40, no. 3, Sep. 2002, doi: 10.1029/2000RG000089.

[123] S. Reyes-Rodríguez, N. Lei, B. Crowgey, L. Udpa, and S. S. Udpa, “Time reversal and microwave techniques for solving inverse problem in Non-Destructive Evaluation,” *NDT & E International*, vol. 62, pp. 106–114, Mar. 2014, doi: 10.1016/j.ndteint.2013.11.003.

[124] J. H. Holland, “Genetic Algorithms,” *Sci Am*, vol. 267, no. 1, pp. 66–73, 1992.

[125] C. Darwin, *On the Origin of Species by Means of Natural Selection, or Preservation of Favoured Races in the Struggle for Life*. 1859.

[126] S. Katoch, S. S. Chauhan, and V. Kumar, “A review on genetic algorithm: past, present, and future,” *Multimed Tools Appl*, vol. 80, no. 5, pp. 8091–8126, Feb. 2021, doi: 10.1007/s11042-020-10139-6.

[127] J. Pereira, J. Mendes, J. S. S. Júnior, C. Viegas, and J. R. Paulo, “A Review of Genetic Algorithm Approaches for Wildfire Spread Prediction Calibration,” *Mathematics*, vol. 10, no. 3, p. 300, Jan. 2022, doi: 10.3390/math10030300.

[128] M. A. Matos, A. M. A. C. Rocha, L. A. Costa, and F. Alvelos, “A Genetic Algorithm for Forest Firefighting Optimization,” 2022, pp. 55–67. doi: 10.1007/978-3-031-10562-3_5.

[129] J. Mandel *et al.*, “Retrieving Fire Perimeters and Ignition Points of Large Wildfires from Satellite Observations,” Mar. 28, 2019. doi: 10.1002/essoar.10500850.1.

[130] B. Shaddy *et al.*, “Generative Algorithms for Fusion of Physics-Based Wildfire Spread Models with Satellite Data for Initializing Wildfire Forecasts,” *Artificial Intelligence for the Earth Systems*, vol. 3, no. 3, Jul. 2024, doi: 10.1175/AIES-D-23-0087.1.

[131] M. Zheng *et al.*, “Conditional Wasserstein generative adversarial network-gradient penalty-based approach to alleviating imbalanced data classification,” *Inf Sci (N Y)*, vol. 512, pp. 1009–1023, Feb. 2020, doi: 10.1016/j.ins.2019.10.014.

[132] J. Mandel, J. D. Beezley, and A. K. Kochanski, “Coupled atmosphere-wildland fire modeling with WRF 3.3 and SFIRE 2011,” *Geosci Model Dev*, vol. 4, no. 3, pp. 591–610, Jul. 2011, doi: 10.5194/gmd-4-591-2011.

[133] J. Haley, A. Farguell Caus, A. K. Kochanski, S. Schranz, and J. Mandel, “Data likelihood of active fires satellite detection and applications toignition estimation and data assimilation,” in *Advances in forest fire research 2018*, Imprensa da Universidade de Coimbra, 2018, pp. 959–968. doi: 10.14195/978-989-26-16-506_105.

[134] A. Farguell Caus, J. Haley, A. K. Kochanski, A. C. Fité, and J. Mandel, “Assimilation of Fire Perimeters and Satellite Detections by Minimization of the Residual in a Fire Spread Model,” 2018, pp. 711–723. doi: 10.1007/978-3-319-93701-4_56.

[135] L. Roques *et al.*, “Emerging strains of watermelon mosaic virus in Southeastern France: model-based estimation of the dates and places of introduction,” *Sci Rep*, vol. 11, no. 1, p. 7058, Mar. 2021, doi: 10.1038/s41598-021-86314-y.

[136] A. Benali *et al.*, “Determining Fire Dates and Locating Ignition Points With Satellite Data,” *Remote Sens (Basel)*, vol. 8, no. 4, p. 326, Apr. 2016, doi: 10.3390/rs8040326.

[137] S. Andrade, N. Shafiei, and P. Mehlitz, “Smoke or cloud: Real-time satellite image segmentation in a wildfire data integration application,” *Comput Geosci*, vol. 204, p. 105960, Oct. 2025, doi: 10.1016/j.cageo.2025.105960.

[138] N. V. Naseri, E. M. Alazzawi, A. A. Hameed, A. Jamil, and F. Soleimani, “Enhancing Forest Fire Ignition Prediction with Data Augmentation and Deep Learning Models,” 2025, pp. 883–894. doi: 10.1007/978-3-031-87154-2_69.

[139] C. Abboud, O. Bonnefon, E. Parent, and S. Soubeyrand, “Dating and localizing an invasion from post-introduction data and a coupled reaction–diffusion–absorption model,” *J Math Biol*, vol. 79, no. 2, pp. 765–789, Jul. 2019, doi: 10.1007/s00285-019-01376-x.

[140] M. Schüle, G. Domes, C. Schwanitz, and T. Heinken, “Early natural tree regeneration after wildfire in a Central European Scots pine forest: Forest management, fire severity and distance matters,” *For Ecol Manage*, vol. 539, p. 120999, Jul. 2023, doi: 10.1016/j.foreco.2023.120999.

[141] ForeFire, “Fuels.ff Fuel Attribute Table.” Accessed: Apr. 02, 2025. [Online]. Available: https://github.com/forefireAPI/firefront/blob/5884afeac7dd91ee4795455da20d0b3e78b603da/examples/aullene/fuels.ff

[142] Windy.app, “Windy.” Accessed: Jan. 25, 2024. [Online]. Available: https://windy.app/

[143] UCAR, “Network Common Data Form (NetCDF).” Accessed: Jan. 25, 2024. [Online]. Available: https://www.unidata.ucar.edu/software/netcdf/

[144] C. Hackett, R. D. A. Moral, and C. Markham, “Simulating Disease in Periods of Low Mobility Using a Hybrid Diffusion and Compartmental Model Built on Geographic Data,” in *2021 32nd Irish Signals and Systems Conference, ISSC 2021*, 2021. doi: 10.1109/ISSC52156.2021.9467871.

[145] S. Fortune, “A sweepline algorithm for Voronoi diagrams,” *Algorithmica*, vol. 2, no. 1–4, pp. 153–174, Nov. 1987, doi: 10.1007/BF01840357.

[146] Yatoom, “foronoi.” Accessed: Jan. 25, 2024. [Online]. Available: https://github.com/Yatoom/foronoi

[147] SNAP, “How to use the SNAP API from Python.” Accessed: Jan. 25, 2024. [Online]. Available: https://senbox.atlassian.net/wiki/spaces/SNAP/pages/19300362/How+to+use+the+SNAP+API+from+Python

[148] I. Al-Rawi, “Implementation of an Efficient Scan-Line Polygon Fill Algorithm ,” *Computer Engineering and Intelligent Systems*, vol. 5, no. 4, pp. 22–28, 2014.

[149] F. A. Albini, “Computer-based models of wildland fire behavior: a user’s manual,” *Ogden, UT: USDA Forest Service, Intermountain Forest and Range Experiment Station. 71 p.*, 1976.

[150] SciPy, “Interpolation (scipy.interpolate).” Accessed: Jan. 25, 2024. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/interpolate.html

[151] Girardi, “Cubic spline for non-monotonic data (not a 1d function).” Accessed: Jan. 25, 2024. [Online]. Available: https://stackoverflow.com/questions/67460967/cubic-spline-for-non-monotonic-data-not-a-1d-function

[152] F. Zehra, M. Javed, D. Khan, and M. Pasha, “Comparative Analysis of C++ and Python in Terms of Memory and Time,” *Preprints (Basel)*, 2020.

[153] K. Słomska-Przech, T. Panecki, and W. Pokojski, “Heat Maps: Perfect Maps for Quick Reading? Comparing Usability of Heat Maps with Different Levels of Generalization,” *ISPRS Int J Geoinf*, vol. 10, no. 8, p. 562, Aug. 2021, doi: 10.3390/ijgi10080562.

[154] A. Farguell, A. Cortés, T. Margalef, J. R. Miró, and J. Mercader, “Scalability of a multi-physics system for forest fire spread prediction in multi-core platforms,” *J Supercomput*, vol. 75, no. 3, pp. 1163–1174, Mar. 2019, doi: 10.1007/s11227-018-2330-9.

[155] A. Trucchia, V. Egorova, A. Butenko, I. Kaur, and G. Pagnini, “RandomFront 2.3: a physical parameterisation of fire spotting for operational fire spread models – implementation in WRF-SFIRE and response analysis with LSFire+,” *Geosci Model Dev*, vol. 12, no. 1, pp. 69–87, Jan. 2019, doi: 10.5194/gmd-12-69-2019.

[156] CORINE, “CORINE Land Cover.” Accessed: Nov. 26, 2025. [Online]. Available: https://land.copernicus.eu/en/products/corine-land-cover

[157] C. Hackett, R. de Andrade Moral, G. Misra, T. McCarthy, and C. Markham, “An efficient method to simulate wildfire propagation using irregular grids,” *Natural Hazards and Earth System Sciences*, vol. 25, no. 8, pp. 2909–2928, Aug. 2025, doi: 10.5194/nhess-25-2909-2025.

[158] Met Éireann, “Historical Data.” Accessed: Feb. 11, 2025. [Online]. Available: https://www.met.ie/climate/available-data/historical-data

[159] S. Sinharay, “Discrete Probability Distributions,” in *International Encyclopedia of Education*, Elsevier, 2010, pp. 132–134. doi: 10.1016/B978-0-08-044894-7.01721-8.

[160] N. Saini, “Review of Selection Methods in Genetic Algorithms,” *International Journal Of Engineering And Computer Science*, vol. 6, no. 12, pp. 22261–22263, 2017.

[161] F. Chicano, A. M. Sutton, L. D. Whitley, and E. Alba, “Fitness Probability Distribution of Bit-Flip Mutation,” *Evol Comput*, vol. 23, no. 2, pp. 217–248, Jun. 2015, doi: 10.1162/EVCO_a_00130.

[162] J. A. Balik *et al.*, “Biogeographic patterns of daily wildfire spread and extremes across North America,” *Frontiers in Forests and Global Change*, vol. 7, May 2024, doi: 10.3389/ffgc.2024.1355361.

[163] H. Ohkura, “Meiosis: an overview of key differences from mitosis.,” *Cold Spring Harb Perspect Biol*, vol. 7, no. 5, Jan. 2015, doi: 10.1101/cshperspect.a015859.

[164] H. Shimodaira, “A diversity-control-oriented genetic algorithm (DCGA): performance in function optimization,” in *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, IEEE, 2001, pp. 44–51. doi: 10.1109/CEC.2001.934369.

[165] A. Abouali, D. X. Viegas, and J. R. Raposo, “Analysis of the wind flow and fire spread dynamics over a sloped–ridgeline hill,” *Combust Flame*, vol. 234, p. 111724, Dec. 2021, doi: 10.1016/j.combustflame.2021.111724.

[166] X. Jin and J. Han, “K-Means Clustering,” in *Encyclopedia of Machine Learning*, Boston, MA: Springer US, 2011, pp. 563–564. doi: 10.1007/978-0-387-30164-8_425.

[167] EFFIS, “EFFIS Wildfire Risk Viewer.” Accessed: Nov. 03, 2025. [Online]. Available: https://forest-fire.emergency.copernicus.eu/apps/fire.risk.viewer/

[168] K. C. Short, “Spatial wildfire occurrence data for the United States, 1992-2020 [FPA_FOD_20221014],” 2022. doi: 10.2737/RDS-2013-0009.6.

[169] C. Hackett, R. de Andrade Moral, D. Rooney, M. Doran, and C. Markham, “FilterPaperIgnitions,” 2025, *Kaggle*. doi: 10.34740/KAGGLE/DSV/12344905.

[170] V. Kumar Goel, A. Chopra, P. Tripathy, and M. Kumar, “Aerodynamics studies of various passenger vehicles using Computational Fluid Dynamics (CFD),” *International Journal of Mechanical Engineering and Technology (IJMET)*, vol. 15, no. 6, pp. 1–9, 2024.

[171] Y. Seo, “Effect of hydraulic diameter of flow straighteners on turbulence intensity in square wind tunnel,” *HVAC&R Res*, vol. 19, no. 2, pp. 141–147, Feb. 2013, doi: 10.1080/10789669.2012.749133.

[172] Fisher Scientific, “FisherbrandTM Grade 601 General Purpose Filter Papers.” Accessed: Mar. 18, 2025. [Online]. Available: https://www.fishersci.co.uk/shop/products/grade-601-general-purpose-filter-papers/11435248

[173] OHAUS©, “PioneerTM Balances Instruction Manual.” Accessed: Dec. 04, 2025. [Online]. Available: https://www.uvm.edu/cosmolab/om/PioneerOHAUSbalance.pdf

[174] T. J. Ohlemiller, “Modeling of smoldering combustion propagation,” *Prog Energy Combust Sci*, vol. 11, no. 4, pp. 277–310, Jan. 1985, doi: 10.1016/0360-1285(85)90004-8.

[175] J. G. Speight, “Combustion of hydrocarbons,” in *Handbook of Industrial Hydrocarbon Processes*, Elsevier, 2020, pp. 421–463. doi: 10.1016/B978-0-12-809923-0.00010-2.

[176] C. Hackett, R. de Andrade Moral, D. Rooney, M. Doran, and C. Markham, “A Computer Vision Technique for the Validation of Wildfire Models using a Desktop Apparatus,” *Proceedings of the Irish Machine Vision and Image Processing Conference 2025 Irish Pattern Recognition and Classification Society*, pp. 10–17, 2025, doi: 10.21251/8043511f-bf93-4b36-9348-0726af0987f6.

[177] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Trans Pattern Anal Mach Intell*, vol. 22, no. 11, pp. 1330–1334, 2000, doi: 10.1109/34.888718.

[178] OpenCV, “Basic concepts of the homography explained with code.” Accessed: Jun. 30, 2025. [Online]. Available: https://docs.opencv.org/4.x/d9/dab/tutorial_homography.html

[179] Robert Fisher, “Computing Plane Projective Transformations.” Accessed: Jun. 30, 2025. [Online]. Available: https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/EPSRC_SSAZ/node11.html

[180] OpenCV, “Operations on arrays”, Accessed: Jun. 29, 2025. [Online]. Available: https://docs.opencv.org/4.x/d2/de8/group__core__array.html#ga6fef31bc8c4071cbc114a758a2b79c14

[181] J. Canny, “A Computational Approach to Edge Detection,” *IEEE Trans Pattern Anal Mach Intell*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986, doi: 10.1109/TPAMI.1986.4767851.

[182] P. Arbelaez, “Boundary Extraction in Natural Images Using Ultrametric Contour Maps,” in *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW’06)*, IEEE, 2006, pp. 182–182. doi: 10.1109/CVPRW.2006.48.

[183] S. Suzuki and K. be, “Topological structural analysis of digitized binary images by border following,” *Comput Vis Graph Image Process*, vol. 30, no. 1, pp. 32–46, Apr. 1985, doi: 10.1016/0734-189X(85)90016-7.

[184] OpenCV, “Structural Analysis and Shape Descriptors.” Accessed: Jul. 01, 2025. [Online]. Available: https://docs.opencv.org/3.4/d3/dc0/group__imgproc__shape.html#ga17ed9f5d79ae97bd4c7cf18403e1689a

[185] OpenCV, “Contours in OpenCV.” Accessed: Jun. 29, 2025. [Online]. Available: https://docs.opencv.org/4.x/d3/d05/tutorial_py_table_of_contents_contours.html

[186] SciPy, “griddata.” Accessed: Jun. 29, 2025. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.griddata.html

[187] U. Krause, M. Schmidt, and C. Lohrer, “A numerical model to simulate smouldering fires in bulk materials and dust deposits,” *J Loss Prev Process Ind*, vol. 19, no. 2–3, pp. 218–226, Mar. 2006, doi: 10.1016/j.jlp.2005.03.005.

[188] R. C. Rowe, P. J. Sheskey, S. C. Owen, and A. P. Association, *Handbook of Pharmaceutical Excipients*. Pharmaceutical Press, 2006. [Online]. Available: https://books.google.ie/books?id=BGJqAAAAMAAJ

[189] Tomar Monika, Ajay Kumar Singh, and Amit Raj Sinha, “Effect of Moisture Content of Exicipient (Microcrystalline Cellulose) on Direct Compressible Solid Dosage Forms,” *Int J Pharm Sci Res*, vol. 8, no. 1, pp. 282–88, 2017.

[190] A. M. Adel, Z. H. Abd El-Wahab, A. A. Ibrahim, and M. T. Al-Shemy, “Characterization of microcrystalline cellulose prepared from lignocellulosic materials. Part II: Physicochemical properties,” *Carbohydr Polym*, vol. 83, no. 2, pp. 676–687, Jan. 2011, doi: 10.1016/j.carbpol.2010.08.039.

[191] Malat’ák J., Velebil J., and Bradna J., “Specialty types of waste paper as an energetic commodity,” *Agronomy Research*, vol. 16, no. 2, pp. 534–542, 2018.

[192] Sartorius AG, “Qualitative & Technical Papers, Creped/ Grade 601/N / ⌀ 240 mm Filter Discs.” Accessed: Mar. 18, 2025. [Online]. Available: https://shop.sartorius.com/ie/p/qualitative-technical-papers-creped-grade-601n--240-mm-filter-discs/FT-3-354-240#

[193] P. L. Andrews, M. G. Cruz, and R. C. Rothermel, “Examination of the wind speed limit function in the Rothermel surface fire spread model,” *Int J Wildland Fire*, vol. 22, no. 7, p. 959, 2013, doi: 10.1071/WF12122.

[194] Missoula Fire Sciences Laboratory, “Communication,” 2024.

[195] C. Hackett, R. de A. Moral, and C. Markham, “A Method to Identify Wildfire Ignition Points and Propagation Durations Using Genetic Algorithms,” Oct. 2024, doi: 10.21203/rs.3.rs-5099486/v1.

[196] Commission for Communications Regulation, “Outdoor Mobile Coverage Map.” Accessed: Apr. 13, 2025. [Online]. Available: https://coveragemap.comreg.ie/map

[197] Irish Aviation Authority, “Safety & Regulations.” Accessed: Jun. 27, 2025. [Online]. Available: https://www.iaa.ie/general-aviation/drones/faqs/safety-regulations

[198] A. G. McArthur, “Fire Behaviour in Eucalypt Forests,” *Leaflet No. 107. Ninth Commonwealth Forestry Conference, India, 1968*, 1968.

[199] Terrain-AI, “Deca-Map Wins Microsoft FHL.” Accessed: Apr. 08, 2025. [Online]. Available: https://terrainai.com/index.php/2023/02/20/deca-map-wins-microsoft-fhl/

[200] Maynooth University, “Tánaiste reveals MU wildfire project as winner of SFI-Defence Challenge.” Accessed: Apr. 08, 2025. [Online]. Available: https://www.maynoothuniversity.ie/national-centre-geocomputation-ncg/news/t-naiste-reveals-mu-wildfire-project-winner-sfi-defence-challenge

[201] European Union Civil Protection Knowledge Network, “Day 3 of EU MODEX IRE EX.” Accessed: Apr. 08, 2025. [Online]. Available: https://civil-protection-knowledge-network.europa.eu/news/day-3-eu-modex-ire-ex

[202] RTÉ News, “Simulation in Wicklow to assess Irish ability to deal with large wildfires.” Accessed: Apr. 08, 2025. [Online]. Available: https://www.rte.ie/news/2025/0326/1504011-forest-fire-exercise/

[203] C. Tymstra, B. J. Stocks, X. Cai, and M. D. Flannigan, “Wildfire management in Canada: Review, challenges and opportunities,” *Progress in Disaster Science*, vol. 5, p. 100045, Jan. 2020, doi: 10.1016/j.pdisas.2019.100045.

[204] IOPscience, “Physics Education.” Accessed: Dec. 22, 2025. [Online]. Available: https://iopscience.iop.org/journal/0031-9120

[205] Sentinel Hub, “EO Browser.” Accessed: Jan. 25, 2024. [Online]. Available: https://www.sentinel-hub.com/explore/eobrowser/

[206] P. S. Chavez, “An improved dark-object subtraction technique for atmospheric scattering correction of multispectral data,” *Remote Sens Environ*, vol. 24, no. 3, pp. 459–479, Apr. 1988, doi: 10.1016/0034-4257(88)90019-3.

[207] QGIS Development Team, “QGIS.” Accessed: Jan. 25, 2024. [Online]. Available: https://qgis.org/en/site/

[208] S. Gilmore, A. Nalband, and A. Dewan, “Effectiveness of DOS (Dark-Object Subtraction) method and water index techniques to map wetlands in a rapidly urbanising megacity with Landsat 8 data,” *CEUR Workshop Proc*, vol. 1323, pp. 100–108, Jan. 2015.

[209] C. Sun, Y. Bian, T. Zhou, and J. Pan, “Using of Multi-Source and Multi-Temporal Remote Sensing Data Improves Crop-Type Mapping in the Subtropical Agriculture Region,” *Sensors*, vol. 19, no. 10, p. 2401, May 2019, doi: 10.3390/s19102401.

[210] T. A. Kebede, B. T. Hailu, and K. V. Suryabhagavan, “Evaluation of spectral built-up indices for impervious surface extraction using Sentinel-2A MSI imageries: A case of Addis Ababa city, Ethiopia,” *Environmental Challenges*, vol. 8, p. 100568, Aug. 2022, doi: 10.1016/j.envc.2022.100568.

[211] S. Sibanda and F. Ahmed, “Modelling historic and future land use/land cover changes and their impact on wetland area in Shashe sub-catchment, Zimbabwe,” *Model Earth Syst Environ*, vol. 7, no. 1, pp. 57–70, Mar. 2021, doi: 10.1007/s40808-020-00963-y.

[212] A. M. Abdi, “Land cover and land use classification performance of machine learning algorithms in a boreal landscape using Sentinel-2 data,” *GIsci Remote Sens*, vol. 57, no. 1, pp. 1–20, Jan. 2020, doi: 10.1080/15481603.2019.1650447.

[213] M. J. Pringle, M. Schmidt, and D. R. Tindall, “Multi-decade, multi-sensor time-series modelling—based on geostatistical concepts—to predict broad groups of crops,” *Remote Sens Environ*, vol. 216, pp. 183–200, Oct. 2018, doi: 10.1016/j.rse.2018.06.046.

[214] L. Baetens, C. Desjardins, and O. Hagolle, “Validation of Copernicus Sentinel-2 Cloud Masks Obtained from MAJA, Sen2Cor, and FMask Processors Using Reference Cloud Masks Generated with a Supervised Active Learning Procedure,” *Remote Sens (Basel)*, vol. 11, no. 4, p. 433, Feb. 2019, doi: 10.3390/rs11040433.