Implementation & Analysis of Stochastic Agent-Based Methods to Model Wildfire Growth

Tyler Jones

A project submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Blake Barker, Chair
Emily Evans
Benjamin Webb

Department of Mathematics

Brigham Young University

ABSTRACT

Implementation & Analysis of Stochastic Agent-Based Methods to Model Wildfire Growth

Tyler Jones
Department of Mathematics, BYU
Master of Science

Modeling wildfires is both an incredibly complex yet impactful problem that has been extensively researched in various fields. Some of the hardest challenges include lack of quality data and the models sensitivity to initial conditions, yet this has not stopped significant advances in the field. The following project presents a background on modeling wildfire, and proposes a novel stochastic agent-based approach. Despite its simplicity, the model exhibits many promising known behaviors of wildfire spread.

Additionally, a framework is proposed utilizing methods of uncertainty quantification and sensitivity analysis to better understand the behavior and performance of any given wildfire model. These methods show what parameters have the largest impact on the behavior of our model, as well as which parameters may be most sensitive to perturbations or errors in input. The methods proposed are flexible in that they can be applied to any wildfire model, not just the proposed agent-based approach.

Lastly, the relationship between our agent-based model and network theory is explored. It is shown that the objective of fire suppression can be accomplished through the use of the max-flow min-cut theorem, providing practical insights into optimal control to minimize fire spread given an environment. Additionally, a short open question is proposed relating the largest connected component in a random graph to the overall behavior of fire spread. Future work related to this project would continue research into the application of network theory in wildfire spread, as well as developing a pipeline to verify and validate our model's performance on known historical wildfires.

Acknowledgements

Thank you to all the great teachers in my life who laid the foundation for me to continue learning. I am also thankful for the support and patience of my wife Allison, letting me run to whiteboards in the middle of dinner or other activities, as well as help me push through discouragement. Lastly, I would not be where I am today without my parents constant guidance and support. Thank you.

Contents

List of Tables

List of Figures

Chapter 1. Introduction & Problem Statement

The work in this project is just a small part of the collaborative work done by the faculty and students in the Mathematical Fire and Industry Research Experience Lab at BYU. Known informally as the Math FIRE group, research focuses in areas such as wildfire risk analysis, risk prediction, and the ecological effects of burn severity. This paper specifically focuses on the progress and methods of wildfire modeling and risk analysis, as well as applying these learning's to optimal control using the principles of network theory.

1.1 Motivations

In the US alone, in the year 2020 wildfires were directly responsible for at least 37 deaths and over $130 Billion in damages [7]. As such, a better understanding of the principles and behavior of fire spread is critical in order to better protect human lives, lower the cost of fire related damage, as well as better protect the environment. With the Bureau of Land Management being responsible for a staggering 245 million acres of land alone [8], this understanding can directly lead to improved policies to mitigate the risk of fires, as well as better fire suppression tactics for those fires that do occur. This is both critical for urban and rural environments, as 88% of fires are human caused [6], showing that improved understanding of how fire spreads in any environment can greatly impact lives saved, cost of damages, and environmental impact.

1.2 Problem Statement

Modeling and understanding fire spread is a difficult dynamic process to simulate due to the sensitivity to initial conditions, uncertainty of model parameters, and frequent lack of high-quality data. However this has not stopped the advancement of research to understand-

ing the properties of fire spread across various fuel types [1] or topological conditions [2]. Additionally, there are many high fidelity wildfire modeling tools currently available such as SPARK [3], FARSITE [4], or Phoenix [5]. As expected, models which take in a high dimension of initial conditions in hopes of more accurate results require exponentially more computing power and take much longer to run. Additionally, complex models may often lose interpretability, which can be prohibitive when trying to coordinate fire suppression efforts with authorities.

With these difficulties in mind, the work in this paper focuses on three main tasks: the development of a simple wildfire modeling software package, methods to understand the behavior of said model, and then exploring applications of optimal control using network theory in the model. The implications of this project are a better understanding of how to construct a wildfire model, a framework to understand the impact of parameters on any given wildfire model, and practical applications relating network theory to fire suppression.

Chapter 2. Model Development

2.1 Past Work & Inspirations

As mentioned in the introduction, there has been much study into the properties of wildfire spread with many models that attempt to simulate the spread of fire through an environment. The current work and study into the models previously mentioned such as SPARK, FARSITE, or Phoenix have been enlightening, and helped guide the direction of this paper when considering the design of a simple system to model fire growth.

Wildfire is an incredibly complex system, dependent on many different variables such as fuel density, humidity, topology, or wind. This complexity means that accurate models must take all these factors into account, and correctly understand the relationship between each

factor as it relates to fire spread. In extreme cases, wildfires can even reach such high temperatures that the ground fire 'crowns' to the tops of trees, causing two separate fires which could act independently [14]. Additionally, even if we could correctly model this and other phenomenon, fire models are extremely sensitive to initial conditions. Meaning, small perturbations in initial data can yield drastically different results. Various methods or techniques have been applied to try and overcome this issue of sensitivity, such as ensemble Kalman filters and data assimilation [15].

With these considerations in mind, a few basic wildfire models were explored which will be described below. These models primarily fall into two categories: partial differential equation based models, and agent based models. By building simple examples of each and taking into account the difficulties mentioned above, a better understanding was reached of how to build a realistic wildfire model, and the different advantageous and disadvantages of each method.

2.2 Partial Differential Equation Models

Many partial differential equation models have been proposed to model fire, of various levels of complexity. Mandel et al. [15] proposed a relatively advanced model based on the amount of fuel (S) and the temperature (T) in the system. The derivation of which was based on the conservation of energy, balance of fuel supply, and the assumed fuel reaction rate. Their model uses 7 different parameters to capture these properties such as thermal diffusivity, cooling laws, and heat transfers. The model is defined as:

$$\begin{aligned}\frac{\partial T}{\partial t} &= \nabla \cdot (k\nabla T) - \vec{v} \cdot \nabla T + A(Se^{\frac{-B}{T-T_a}} - C(T - T_a)) \\ \frac{\partial S}{\partial t} &= -C_s S e^{\frac{-B}{T-T_a}}, \quad T > T_a,\end{aligned}$$

with the seven parameters being defined in the table below.

Table 2.1: PDE Model Parameters

Parameter	Units	Description
k	$m^2 s^{-1}$	thermal diffusivity
A	$K s^{-1}$	temperature rise per second
B	K	proportionality coefficient in modified Arrhenius law
C	K^{-1}	scaled coefficient of heat transfer to environment
C_s	s^{-1}	fuel relative to disappearance rate
T_a	K	ambient temperature
$\vec{v}$	$m s^{-1}$	wind speed

This model can be solved using finite difference methods, however the derivation of such is outside the scope of this project. However, some advantages of a complex model like this are that it aims to capture the majority of the interactions in an environment during a fire. By taking into account both fuel load and temperature, the model provides a richer understanding of the behavior of potential fires. However, remembering the sensitivity of models to initial conditions, having more parameters means that slight perturbations from true values could vastly impact the performance of any model. To account for this, in every case of which the author is aware, Mandel et al. and others simply calibrate the given coefficients from physically observable quantities of known fires in order to obtain realistic parameterizations for each variable.

In conclusion, models based on partial differential equations have many advantages. Being based in physical laws, these models can be easy to interpret on paper, and can be simple to describe from a 'top down' viewpoint. And while stochastic partial differential equations do exist, these can be quite difficult to solve. Hence while finite element or finite difference methods do exist to approximate solutions for PDE based models, agent-based modeling is considerably easier to implement in practice which was a motivating interest for study in this current work. Additionally, agent-based models easily allow the addition of stochastic

behavior and still provide complex emergent behavior, even when given simplistic interaction criteria. A such, this project focuses on understanding agent-based models, and our design of a realistic wildfire agent-based method.

2.3 Agent-Based Models

When considering other modeling frameworks, agent-based modeling is also a popular choice to model physical phenomenon. Agent-based models focus on describing the behavior of a system through the perspective of its constituent units [10]. As such, it is a flexible framework that requires only three things: an environment, agents, and interaction criteria. Agents can be abstracted to anything capable of carrying a state, which changes based on the stated interaction criteria. These decisions can impact itself, other agents, or the environment in which it resides. This flexible framework makes it ideal for modeling many different situations such as immune response to bacterial pathogens [11], social sciences [12], and wildfires [9].

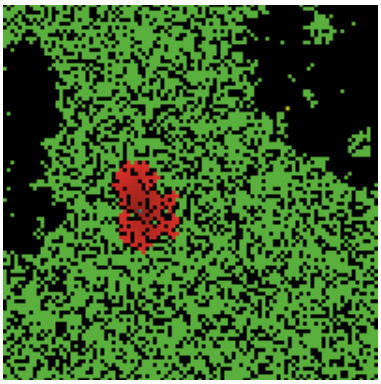

Figure 2.1: Simple NetLogo wildfire simulation

When considering agent-based models and wildfires, a well known software package is the *NetLogo* multi-agent programmable environment. This is a language that allows the user to create various simple agent-based models, and has a well known simple cellular wildfire model [16]. In what could be considered the most basic of agent-based fire models, we have two types of agents, soil or fuel, spread randomly over our two dimensional surface which forms the environment. The only interaction criteria is any cell currently on fire will catch cells to the north, south, east, or west on fire in the next time step.

Despite its simplistic nature, it has been shown that that this model can produce realistic

burn areas of known fires [9]. That being said, some of the weaknesses in agent-based models is that these methods rely on optimizing parameters given known fires, rather than being able to derive said constants through their physical laws. As such its common to face a 'guess and check' process to validate that the model is performing as expected. However these models are easy to interpret, and despite being built from simple agent to agent interactions, can exhibit complex macro-level behavior and are a popular choice for modeling many different dynamic systems.

2.4 Our Model

For this project, the decision was made to create an Agent Based Model, for the reasons of explainability and ease of use in introducing complex interactions between confounding variables. Rather than deriving a potentially complex PDE to capture the effects of wind, topology, fuel, and temperature and having to try and solve this numerically through a finite difference scheme, we can focus on building what we believe the agent to agent based interactions would look like, and aim to understand the behavior of the resulting system. While this does mean our model is not based on physical laws and would need to be validated by seeing its performance on known wildfires, we believe our system has many desirable properties that one would hope to find in a wildfire model.

Additionally, the purposeful choice was made to introduce stochastic elements into our model. When considering wildfires over hundreds of thousands of acres, its impossible to know the exact parameters such as the fuel density load or the burn area over time with exactness. As such, our model focuses on building our environments from assumed distributions, and running many simulations to draw conclusions on the behavior of the given fire. Specifically we created a stochastic wildfire model which takes into account fuel, topology, wind, and temperature. Our model has 10 parameters which are described in the table below:

Table 2.2: Model Parameters

Parameter	Symbol	Range	Description
Tree Distribution	$\mathcal{N}$	-	The distribution of fuel among trees
Mean Tree Fuel	μ^t	[50,200]	Average fuel in each tree
Deviation Tree Fuel	σ^t	[10,100]	Standard deviation of fuel in each tree
Tree Burn Likelihood	θ^t	[.05,.55]	How likely a tree is to burn when introduced to fire
Vegetation Distribution	$\mathcal{N}$	-	The distribution of fuel among vegetation
Mean Vegetation Fuel	μ^v	[5,45]	Average fuel in vegetation
Deviation Vegetation Fuel	σ^v	[1,21]	Standard deviation of fuel in vegetation
Vegetation Burn Likelihood	θ^v	[.4,.9]	How likely vegetation is to burn when introduced to fire
Heat Impact	λ_1	[.1,1]	Tuning parameter for the heat generated by fuel burn
Macro Topology Impact	λ_2	[10,50]	Tuning parameter for the environments topology impact on fire spreading
Micro Topology Impact	λ_3	[.05,1]	Tuning parameter for the environments topology impact on fire spreading
Wind Impact	λ_4	[1,10]	Tuning parameter for the winds impact on fire spreading

In our agent based model, we have 4 total agents: water, soil, vegetation, and trees. Together the union of all of these agents make up our environment, with each 'cell' being a singular agent. In practice, one would take satellite images and other data maps to build an environment, perhaps with some kind of majority based classifier to determine the most

appropriate agent type for each pixel. Additionally we take into account the topology of the environment, which is a property of the agents location in the environment.

Finally in every simulation of the environment, the amount of fuel for each agent is pulled from the respective distribution as given in our table of parameters, which in its current implimentation is assumed to be normal. This is because, as mentioned previously, getting accurate fuel maps is extremely difficult, and rather than assume that the amount of fuel for each agent class is uniform we work with a distribution of fuel that is identical and independently distributed for each agent. A breakdown of these different layers of an environment is seen below, where both fuel and topology layers are blue for lower values of fuel or elevation, and red for high values.

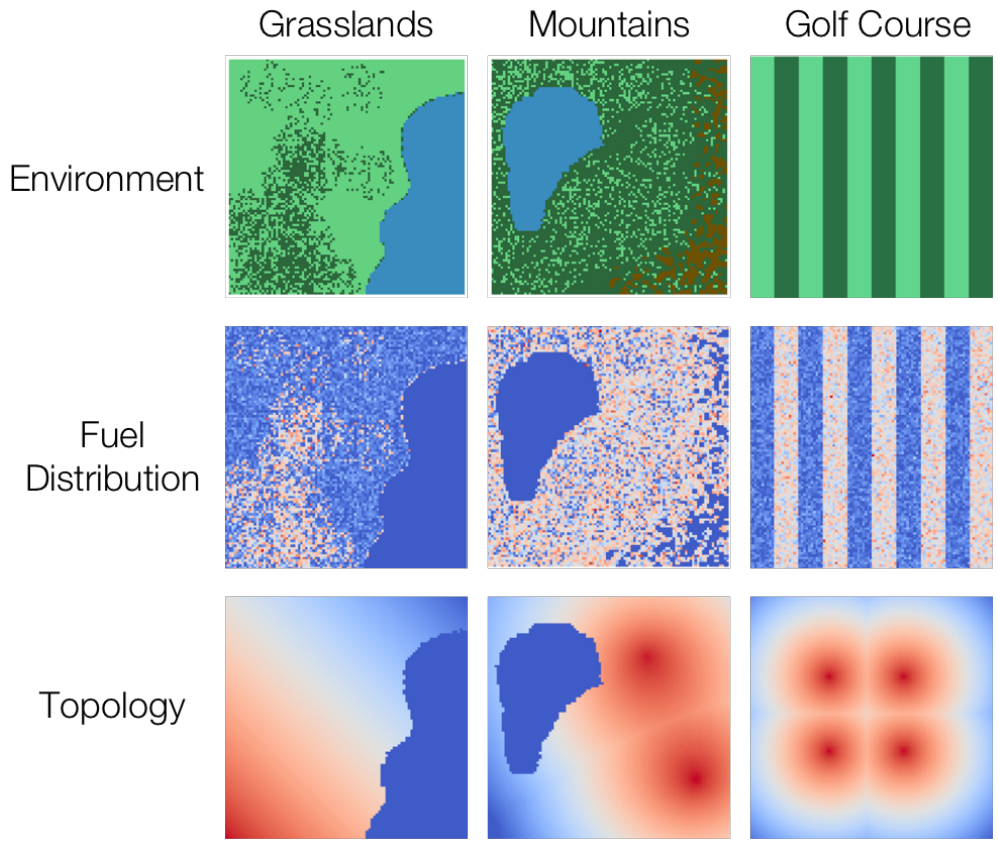

Figure 2.2: Three test environments and their respective layers

All future work with the model was done with these three test environments. In each case,

they were developed to be uniquely different in their properties, in order to better understand how the model may perform under various circumstances. The first environment, *Grasslands*, has a gentle slope in elevation with plentiful low density fuel that is quick to catch fire, meant to resemble and mimic grassland fires. The second environment is *Mountains*, which has densely populated trees with a high fuel load, along with an aggressive slope forming into the peaks of two mountains. Lastly as an edge case, we have the *Golf Course* environment that has stripes of low and high fuels, along with 4 equidistant hills in the middle. By running identical tests on all three environments, we can compare the similarities and differences across each and better understand model behavior across a spectrum of circumstances.

Now that we've established our agents, environment, and their properties, we move on to the explicit definition of our agent based model. We first list the three main rules governing the evolution of fire through time. These are as follows:

(i) **Agent State:** An agent carries one of three states: no fire, fire, or burned out

$$s_{i,j}^t \in \{NF, F, BO\}$$

(ii) **Stability:** An agent which is burned out, will never catch fire again

$$s_{i,j}^t = BO \implies s_{i,j}^{t+k} = BO \quad \forall k \in N$$

(iii) **Probability of Burning:** The probability of catching fire is a function of an agents own properties, and that of each neighbor

$$p_{i,j}^{t+1} = f(N(i,j), a_{i,j})$$

The actual fire transition probability function is built off of the *sigmoid* function. When deciding upon the fire transition function, the sigmoid function was chosen because it lends itself nicely to probabilities as its range is between 0 and 1. Hence for a given cell, the probability of it catching on fire on the next time step is given as:

$$p_{i,j}^{t+1} = \delta_{f_{i,j}^t} \delta_{N_f} \text{sigmoid}(\theta_{i,j} + s_{i,j}^t + t_{i,j} + w_{i,j}^t) \tag{2.1}$$

Where $\delta_{f_{i,j}^t}$ and δ_{N_f} are the indicator functions for if the agent $a_{i,j}$ is flammable at time t, and if the neighborhood of $a_{i,j}$ contains a cell on fire. If both functions are positive, then it means our agent is flammable, and is currently bordered by an actively burning cell. Each component of our sigmoid function is defined in the table below. For every use of Δ we are taking the difference of some attribute between the agent $a_{i,j}$ and the cell on fire, $\hat{a}$. As such $\Delta\mu_{i,j} = \mu_{i,j} - \hat{\mu}$, the difference of fuel between our agent and the cell on fire, and $\Delta v_{i,j} = v_{i,j} - \hat{v}$ is the difference in elevation between our agent and the cell on fire. The vector $\vec{x}$ represents the position vector of the flammable cell relative to our agent.

Table 2.3: Model Components

Symbol	Component	Definition	Description
$\theta_{i,j}$	Fuel Characteristic	$\theta_{i,j} \in [0,1]$	How likely is this fuel type to catch fire?
$s_{i,j}^t$	Heat Characteristic	$\lambda_1\sqrt{\Delta\mu_{i,j}}\text{sign}\Delta\mu_{i,j}$	How much fuel is burning relative to the current cell?
$t_{i,j}$	Topology	$\lambda_2\sin(\lambda_3\tan^{-1}(\Delta_{v_{i,j}}))$	Is the fire traveling up or downhill?
$w_{i,j}^t$	Wind	$\lambda_4\frac{\|\|\vec{w^t}\|\|<\vec{w^t},\vec{x}>}{\|\|<\vec{w^t},\vec{x}>\|\|}$	How much of the wind is pushing parallel to the fire relative to the cell?

Now for the most simplistic of fire simulations seen from *NetLogo* a 4-point neighborhood is used such that $N_4(i,j) = \{(i+1,j),(i-1,j),(i,j-1),(i,j+1)\}$. In development of our model we see much more realistic performance from the 8-point neighborhood, $N_8(i,j) = N_4(i,j)\bigcup\{(i+1,j+1),(i-1,j+1),(i-1,j-1),(i+1,j+1)\}$. Notice that for an initial circular burn, the deterministic 4-neighborhood creates an unrealistic straight line expansion from the point of ignition. Compare this to the 8-point stochastic performance, which maintains the expected circular shape of fire spreading outward, despite having the

square neighborhood N_8 when calculating spread.

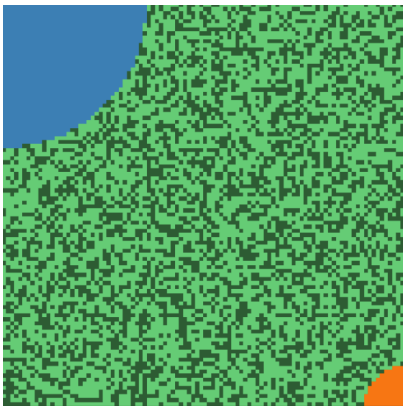

Figure 2.3: Initial fire in test environment

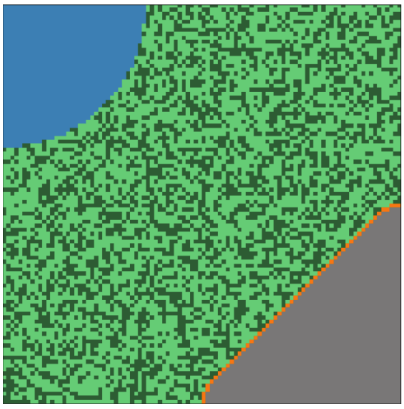

Figure 2.4: Deterministic N_4 model forecast

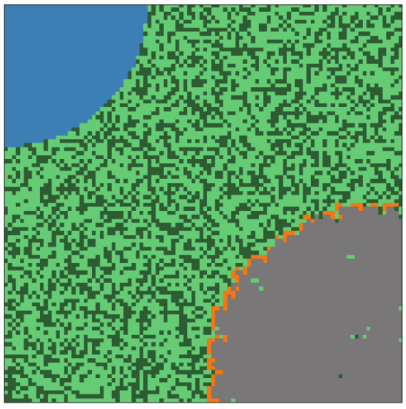

Figure 2.5: Stochastic N_8 model forecast

Additionally, our model accounts for much more sophisticated situations of elevation, fuel differences, and wind conditions compared to the simple 4-point NetLogo model. Recall that the Mountains environment featured two distinct peaks. In our model when starting a fire at the top of the first peak with two different wind directions, we see realistic behavior given wind conditions and elevation.

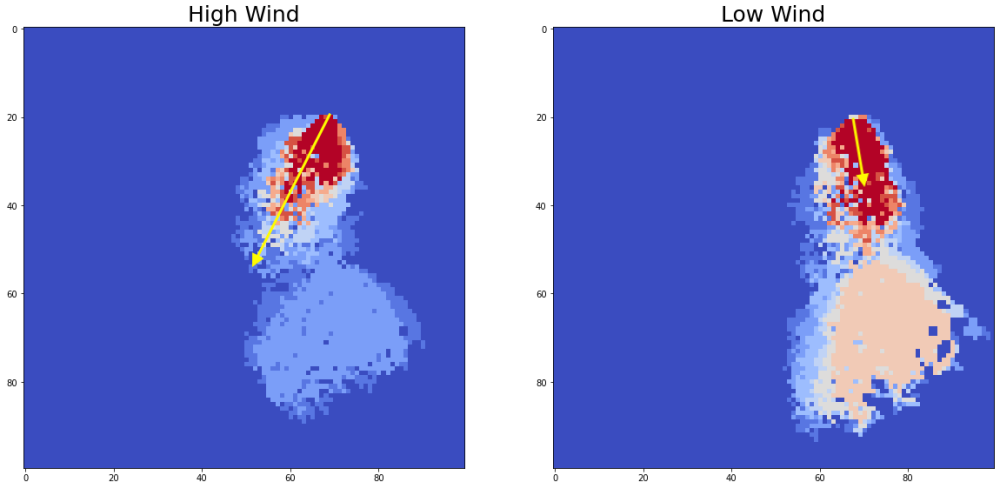

Figure 2.6: The impact of wind and elevation in our model

On the left, when starting at the top of the hill with a high wind vector pointing away from the other peak, very few of the simulations have the fire spread to the base and then up the other peak. This could be attributed to the fact that the wind has to 'push' the fire

downhill, and away from the other peak. However on the right when given a smaller wind vector, but towards the direction of the next peak, there is a much higher probability that the fire spreads and completely encapsulates the second peak.

While these results do not guarantee that our model is accurate, they do provide strong empirical evidence that the behaviors seen in our model mimic those found in real world fires. As such, equation (2.1) may not be perfect, but it does provide a solid mix of stochastic modeling, interpretability, and promising emergent behavior.

Chapter 3. Uncertainty Quantification & Sensitivity Analysis

3.1 Motivations

As mentioned previously, it is important not just to build a good model, but to be able to quantify and understand the behavior of the chosen modeling approach. When considering the complexities of modeling wildfires, the principles of uncertainty quantification and sensitivity analysis are valuable tools to understand the potential strengths and weaknesses of a model. As such these principles can be used regardless of the choice of model, however we will look at them through the lens of our model as described in chapter 2. Through our results, we see how these methods can be used to greater understand the properties of our stochastic agent-based model.

First, it is important to understand the distinction between uncertainty quantification and sensitivity analysis. The former deals with understanding the uncertainty in model output due to uncertainty of the input parameters, where the latter focuses on understanding the contribution of each input parameter on the uncertainty of the output [17]. In the scope of this paper, the model output focused on, sometimes referred to as the *Quantity of Interest*

or QoI, is the total area burned in our model over a specific time frame.

In this paper we propose two methods. The first is the normalized difference method, which is an uncertainty quantification method. Insights generated from these experiments help understand which elements of our model are most sensitive to changes in model output, given uncertainty on their input. These issues frequently occur in not knowing the perfect fuel distribution given an environment, or other respective parameters such as heat transfer coefficients, etc. The second method is Cotter's method, a sensitivity analysis method that ranks model parameters relative to their impact on model output. Knowing what model parameters impact model outputs more than others can help with building faster, lower dimensional surrogate models, or provide motivation for knowing which parameters could take higher priority in getting accurate inputs.

In both of these methods, we run tests on our three test environments, on three different ignition conditions. These conditions are a lightening strike, centered square, and south to north fire. A lightening strike is represented by the single cell caught on fire, and it is common in simulations for the individual agent to burn out without the fire spreading, mimicking realistic behavior. The centered square is meant to emulate a small fire that is isolated to a single area initially, with the south to north fire being an established fire perimeter that is entering into one complete side of our environment.

With the multiple environments and initial conditions for the fire, we have a thorough test bed to understand the behavior of our model under a variety of circumstances. Additionally, the following tests and process could be replicated regardless of our model, making it especially attractive to pursue as it could increase understanding the behavior of any future model developed by the Math FIRE group.

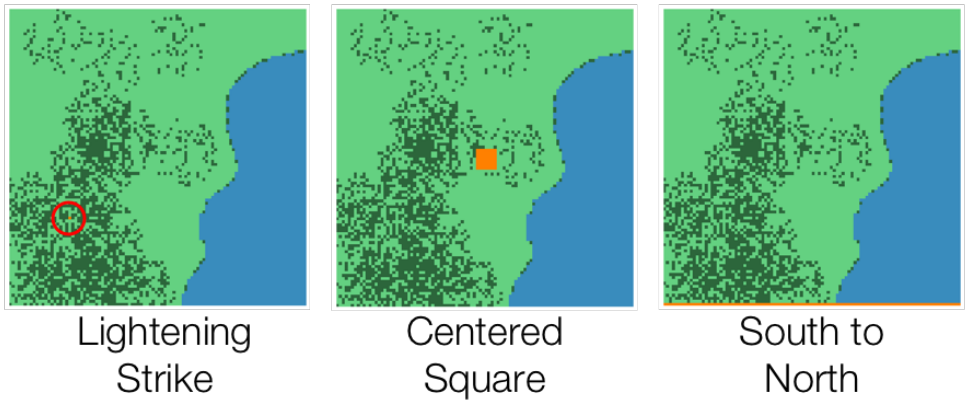

Figure 3.1: Three different initial conditions of fire for testing

3.2 Normalized Difference Method

When modeling wildfires, it is not unreasonable to assume that we will have errors in our parameter estimation. Parameters such as fuel load, or correct initial conditions with fire parameters can be especially troublesome, as keeping an accurate inventory of millions of acres of land is simply infeasible. As such, a method to understand the impact of these errors is especially helpful in understanding which parameters are most sensitive to perturbations in their input. This knowledge then helps us understand which parameters are most important to estimate as closely as possible, and which have minimal impact when perturbed, and hence are more stable.

The proposed method of accomplishing this task is the normalized difference method. The purpose of this method is to define a specific model parameter state x_0, which can be thought of as the current belief of the parameter state of our system. Then for each parameter i we define it's normalized difference measure as:

$$\rho_i = \frac{\partial Q}{\partial x_i}\left(\frac{x_i}{Q_0}\right) \approx \frac{Q_i - Q_0}{x_i - x_{0,i}}\left(\frac{x_i}{Q_0}\right) \tag{3.1}$$

Recalling that our quantity of interest Q is the total area burned in our system, we see that

the normalized difference measure is simply the partial derivative of Q at x_0 with respect to i. While its almost certainly the case that our parameters have a non-linear relationship with the total amount of fire burned in our system, we can use the magnitude of the normalized difference to build the estimated linear effects relative to x_0. Hence $|\rho_j| > |\rho_k|$ would imply that parameter j has a higher impact when perturbed on fire spread, than that of parameter k. Similarly you could rank parameters on importance, or by some threshold τ determine that if $|\rho_k| < \tau$ then the parameter is of minimal importance to the system, and perhaps could be dropped from the model to simplify computation.

In practice, we use the approximation of the normalized difference method as we cannot compute this measure analytically given our system. When computing this numerically in our experiments, x_i represents the parameter state vector x_0, with the i'th entry perturbed by 5% relative to its valid domain, with Q_i as the total area burned given the parameter state vector x_i. With the stochastic nature of our simulations, we run 50 trials for each parameter state vector x_i, and take the median of the resulting outputs for Q_i. Across all ignition points, and all environments, this lead to a total of $4,950$ simulations.

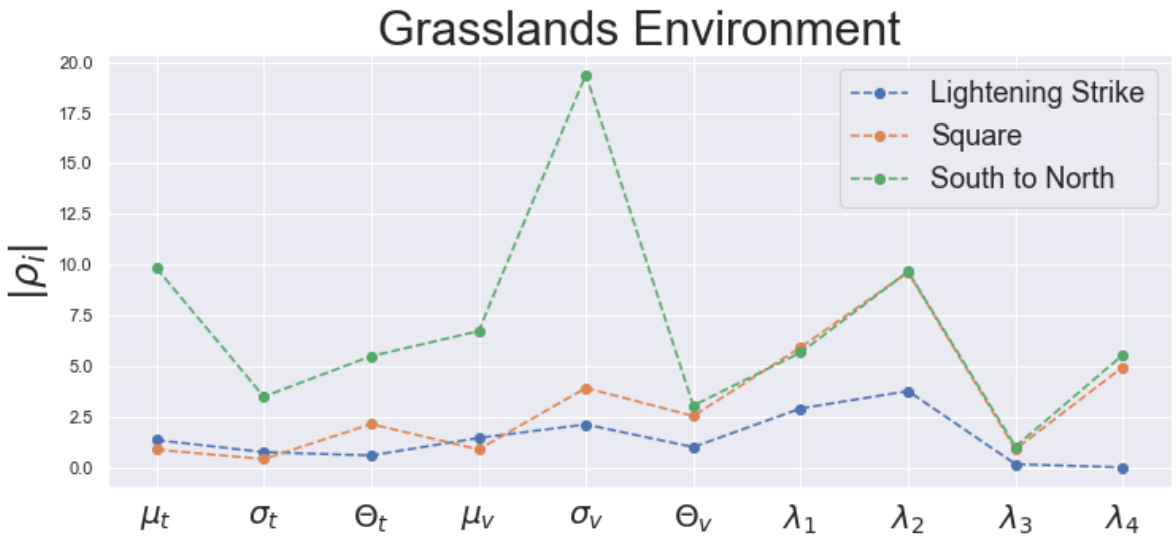

Figure 3.2: Normalized difference for the grasslands environment

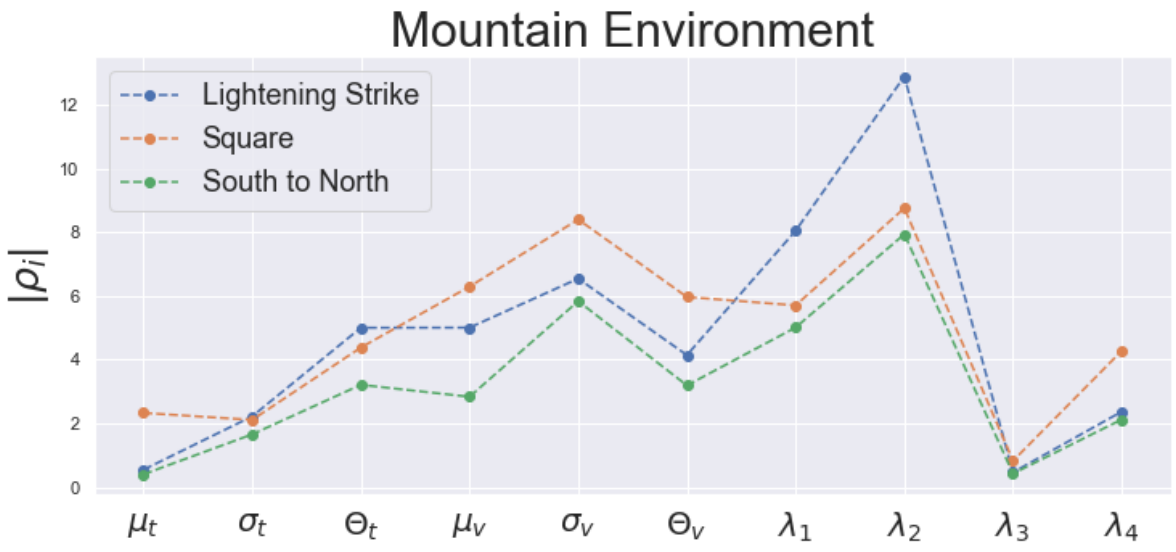

Figure 3.3: Normalized difference for the mountain environment

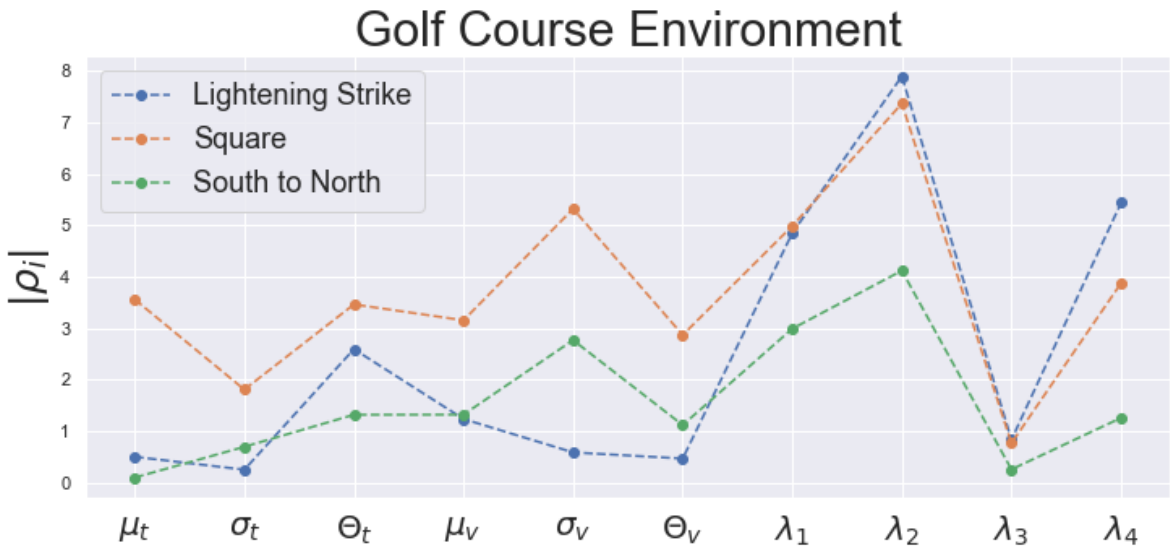

Figure 3.4: Normalized difference for the golf course environment

When looking at our results, we see that in general, there is a rough agreement of ordering in the magnitude of the normalized sensitivity indices across each ignition point, however not across different environments. Note that the in the Grasslands test, σ_v or λ_2 are the parameters with greatest magnitude. This could be explained by noting that Grassland fuel is primarily vegetation, and that there is a sharp slope to the environment. Hence it supports the idea that it is important to have a strong sense of the distribution of the primary fuel load in an environment, as well as the topology of an area if its particularly non-uniform.

We also see that the behavior of the Mountain and Golf Course environment differ from that of the Grasslands, in the fact that the agent based fuel distribution and propensity to burn (θ values) parameters are increasingly more important, as seen from the higher magnitude. This would suggest that slight perturbations in parameters relating to low fuel agents quick to burn is much more stable than compared to those agents or environments with higher fuel loads.

While preliminary, these results give solid insights into the behavior of our model, and could be replicated for any other given system. Important to note as well is that these results focus on the quantity of interest, total fire burned in our system. When considering more advanced quantities such as fire perimeters or intersection over the union of known fires, these measurements could give completely different ordering of magnitudes. However, the normalized difference method provides a powerful way to understand the sensitivity of our parameters to perturbations, to help better understand model behavior.

3.3 Cotters Method

Cotters method is a systematic fractional replicate design pattern, meant to isolate the most important parameters in experiments involving a large number of parameters and interactions [20]. While the normalized difference method focus on the effect of small changes

in parameter values, Cotters method in contrast focuses on a two-level fractional factorial design pattern. In simulations, two runs are dedicated to all parameters being set to their low and high extreme values, followed by $2k$ runs where one parameter is set to one extreme, with all others being assigned to the other extreme. As such, the total number of simulations is $2k + 2$.

We perform the first run t_0 with all parameters set to low, then $t_1, \ldots, t_k$ with everything low, and one parameter high. Then runs $t_{k+1}, \ldots, t_{2k}$ are with all parameters high and one parameter low, finishing with run t_{2k+1} which has all parameters set high. Then we may define the magnitude, M_j as the effect of factor j as follows:

$$\begin{aligned} M(j) &= |C_o(j)| + |C_e(j)| \\ C_o(j) &= \frac{1}{4}\{(y_{2k+1} - y_{k+j}) + (y_j - y_0)\} \\ C_e(j) &= \frac{1}{4}\{(y_{2k+1} - y_{k+j}) - (y_j - y_0)\} \end{aligned} \tag{3.2}$$

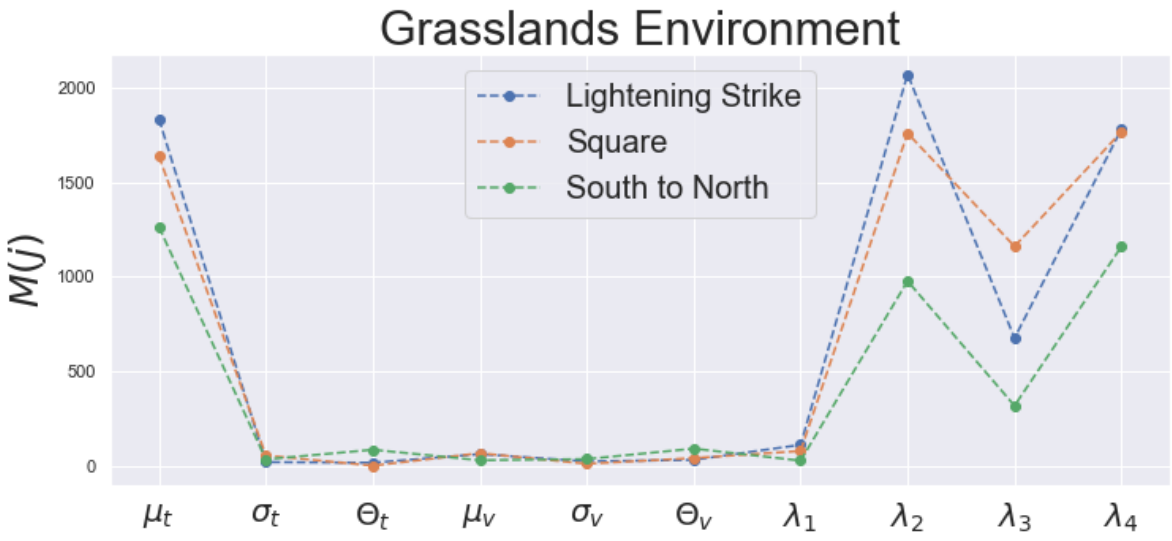

Figure 3.5: Cotters method for the grasslands environment

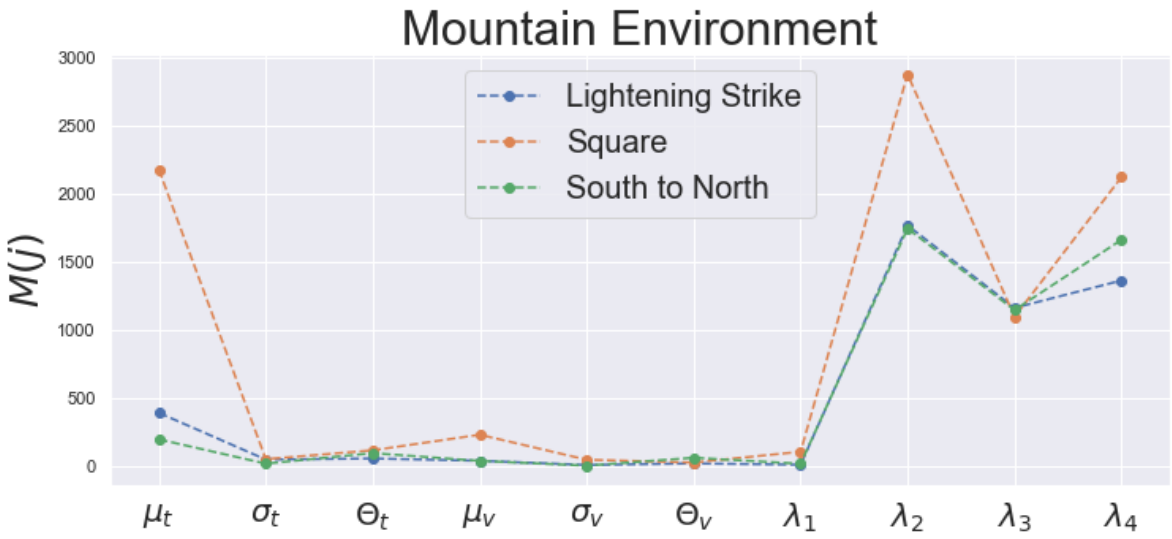

Figure 3.6: Cotters method for the mountain environment

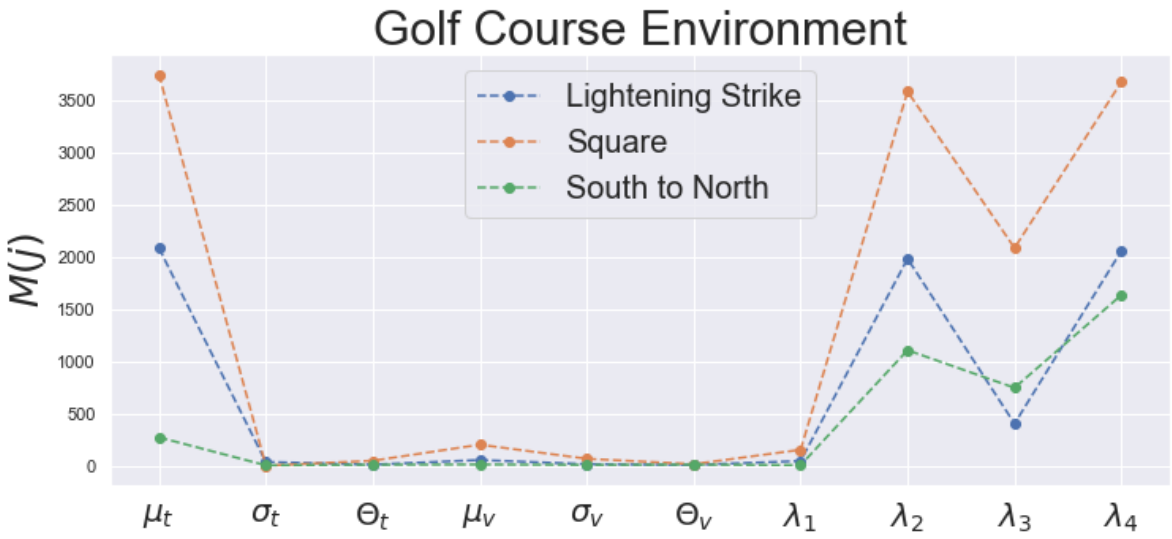

Figure 3.7: Cotters method for the golf course environment

The parameters which have the highest magnitude $M(j)$ are those with greater impact in the output of our model, with C_o and C_e commonly being refereed to as the odd and even order effects of the given parameter. Again due to the stochastic nature of our model, we run the $2k + 2$ simulations 50 times, across each environment and ignition point, yielding 9,900 total runs.

Observing the results for each environment, we see that across all environments and ignition points the ordering of the magnitudes are relatively constant, with $\mu_t, \lambda_2, \lambda_3$ and λ_4 having the highest impact, and all other parameters having negligible magnitude values. These results seems to suggest that the parameters impact, or lack thereof, on model output, is similar regardless of environment makeup or ignition location. This stability is a considerably desirable trait for a model, suggesting that parameter importance is stable across a wide range of environments and ignition points.

Chapter 4. Optimal Control

4.1 Application to Network Theory

Studying wildfire as a process of heat spreading across a network was introduced as the well known "Firefighter Problem" proposed in 1995 [18]. Furthermore, the idea of representing fire spread on a environment as a graph translates perfectly into the idea of agent based models. Given our environment, we can consider each agent as a node in the graph, with connecting edges given by its neighbors. As such in the N_4 scheme, the degree of every node would be 4 with the exception of our boundary, and similarly in the current N_8 model, each non-boundary cell has degree 8. Hence we can view the spread

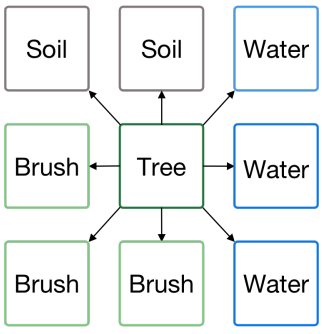

Figure 4.1: Example of interior agent node with its neighbors

of fire on our graph as an ever growing collection of random walks throughout the nodes of our graph, determined by the probability defined in equation (2.1). This allows us to utilize the many tools of network theory in understanding our system further. Of these many applications, we will focus on the use of the max-flow min-cut theorem and how it relates to the optimal control in minimizing fire spread.

4.2 Max Flow - Min Cut Theorem

Given some graph $G = (V, E)$, we may denote an arbitrary node $s \in V$ as the source node, and a node $t \in V$ as the sink node. The max-flow min-cut theorem states that the maximum flow between nodes s to t is equal to the minimum cost of cutting edges such that s and t lie in two separate components. We now describe how we can use this powerful tool to explore optimal fire suppression methods to protect a given sink cell, or node, in our graph.

First, consider the current state of fire spread in our model. Each cell on fire has the potential to ignite each neighboring cell, with some probability. However we now introduce the terminology of a *greedy fire walk*. Rather than a cell spreading fire to any of 8 possible neighbors, it only may pass it to the neighbor with the highest probability to catch fire. A small but important detail of note is the inversion of logic given in our initial model. In equation 2.1, a cell receives fire dependent on each one of its neighbors. However with the greedy fire walk, the cells that are on fire send and ignite only the one cell with the highest probability to catch fire as given in equation 2.1. This distinction is only important in the implementation of the code, as analytically the likelihood of a cell to catch fire remains the same in either case.

Now we define our ignition point, and label it our source node s, and determine some single cell or agent we wish to protect, and denote it as the sink node t. We then forecast a greedy fire walk some arbitrary number of steps, forming a path of the highest probability for fire

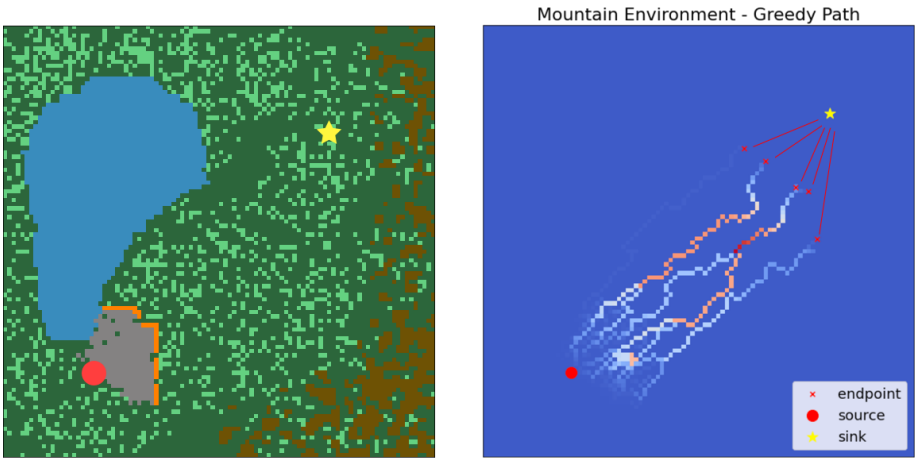

Figure 4.2: Environment with source and sink, with respective greedy walk

spread. This creates a graph from our source to sink node. For every iteration of our greedy walk, if the new cell, or agent, is not in the graph it is added and the edge drawn, however if the node is already in the graph, its edge weight is increased by one. At the end of the forecast we connect all tails of the walk to the sink node with an edge, weighting the edge by the L_2 distance to the sink node. We can see this process in figure 4.2 which shows that the forecast identifies 5 primary fire paths from source to sink, with the colors representing the respective weight of each edge.

Given our graph, with each node representing a cell most likely to catch fire given its neighbors, and edges weighted with the amount of greedy paths connecting the nodes, we can now apply the max-flow min-cut theorem and find the edges with minimal weights that will cut this graph in two. This set of edges $\bar{E}$, represent the safest places to try and separate the graph into two connected components, thus ensuring the flow from the source node never reaches the sink node. The principle of "most safe" comes from the fact that this minimizes the weight of edges cut, meaning that it is guaranteed to be the locations that separate the graph, yet still have minimal fire predicted to travel through the regions. See figure 4.3 for an example result, noticing that 3 of the 4 optimal cuts are placed at the base of the 2 elevated peaks in the environment.

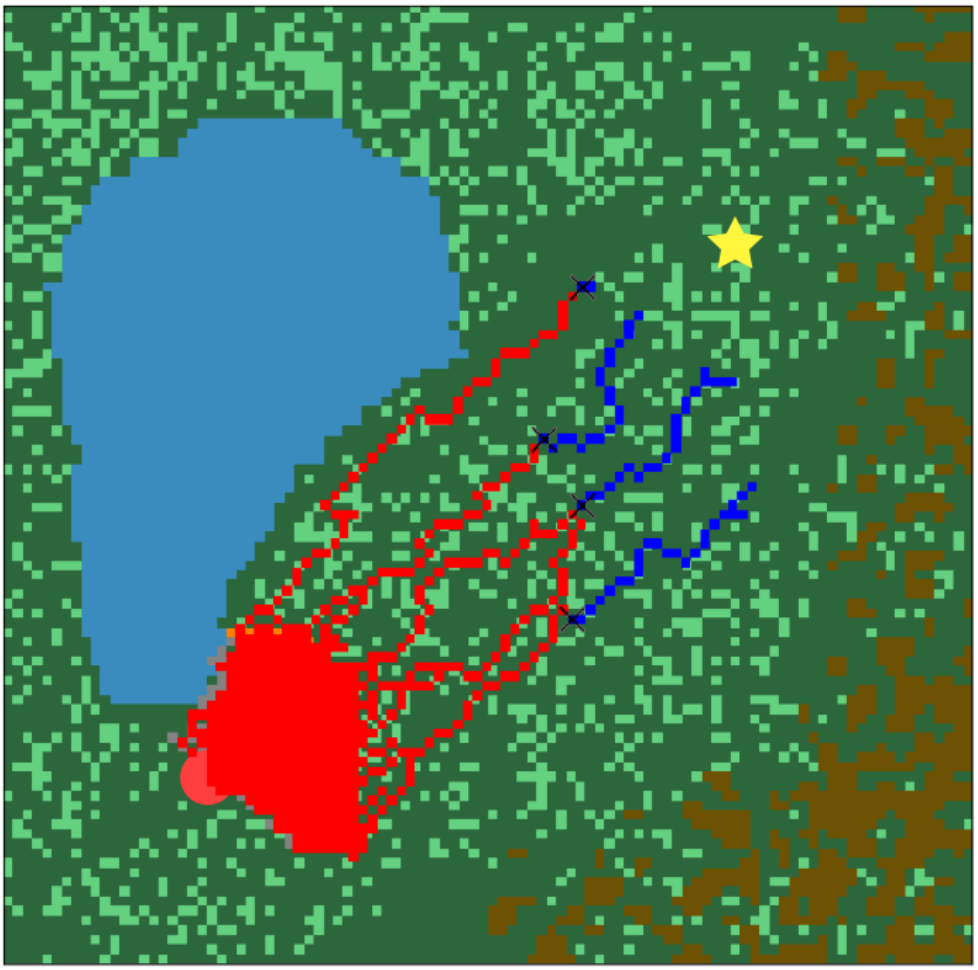

Figure 4.3: Location of optimal cuts to separate the source node from sink

4.3 Random Graphs & Their Application

Lastly there is one area of particular interest to the author, as it relates to random graphs and their connectivity. To introduce the motivation, it is not unreasonable to think that with the exception of spotting, where flying embers ignite fuel farther ahead of the current fire line, fire will most likely travel to its adjacent neighbors. Hence without the intervention of outside sources, and when using the concept of fire on a graph, fire will naturally spread to its entire connected component of which it is a part of, regardless of its initial conditions.

The simple, yet popular deterministic N_4 fire spread model, helps us see this interesting behavior. In this model it is well known that when creating random environments of various

densities of fuel, the relationship between the amount of forest burned and the density of fuel, is extremely non-linear when you ignite one side of the environment. Hence when considering random environments of various fuel density, the problem appears to be identical to considering the size of the largest connected component in the respective graph.

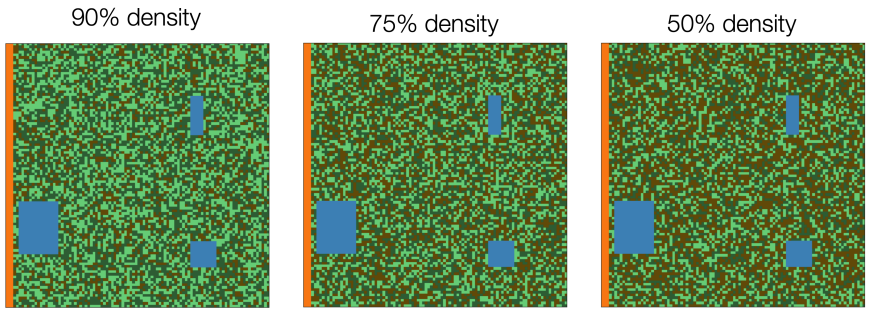

Figure 4.4: Random environment with various levels of fuel density

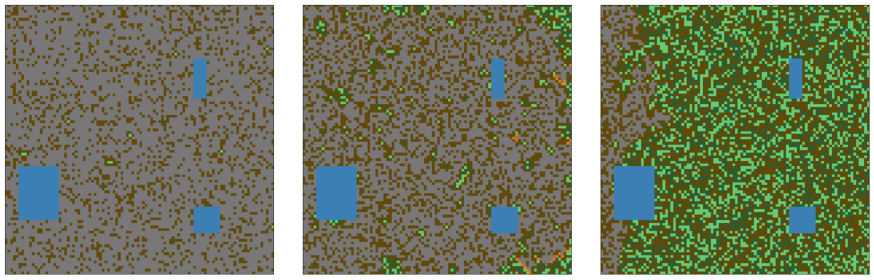

Figure 4.5: Results of fire burn using deterministic N_4 model

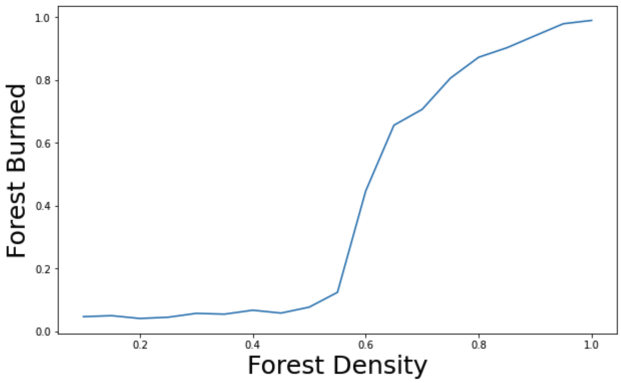

Figure 4.6: Nonlinear relationship between density and total burn

Considering the well known Erdős–Rényi random graph, where of n nodes, an edge between each distinct pair of nodes is randomly placed with probability p, the size of the largest

giant component in this graph, defined S, follows the relationship: $S = 1 - e^{-p(n-1)S}$. While one cannot explicitly solve for S directly, it can be graphically plotted to find solutions [19]. Similarly, we could consider a lattice network in two dimensions to represent our environment, and the size of its largest component as a function of the forest or node density. Hence, the largest connected component would then dominate the total area burned in the environment.

Thus the total area burned in an environment with a specific density of fuel in a deterministic model is thought to be directly related to the largest connected component of a trimmed lattice graph, where trimming involves removing random nodes and their edges to reach the same density as the environment. While no analytic conclusions were reached proving this relationship or the expected size of the largest connected component, empirical results found that total area burned to be highly correlated to the largest connected component in the corresponding 'trimmed lattice'.

In the experiments, a given lattice L was created which was the same size as the environment containing N total nodes. Then for each given density d, a sub-graph L_d was created where $(1-d)N$ nodes were randomly removed from the graph, leaving behind the trimmed lattice, L_d. For each density d, this process was repeated 100 times creating random sub-graphs and the median size of the largest connected component for each sub-graph L_d was taken. The resulting curve, and its relation to the deterministic N_4 model, is shown below.

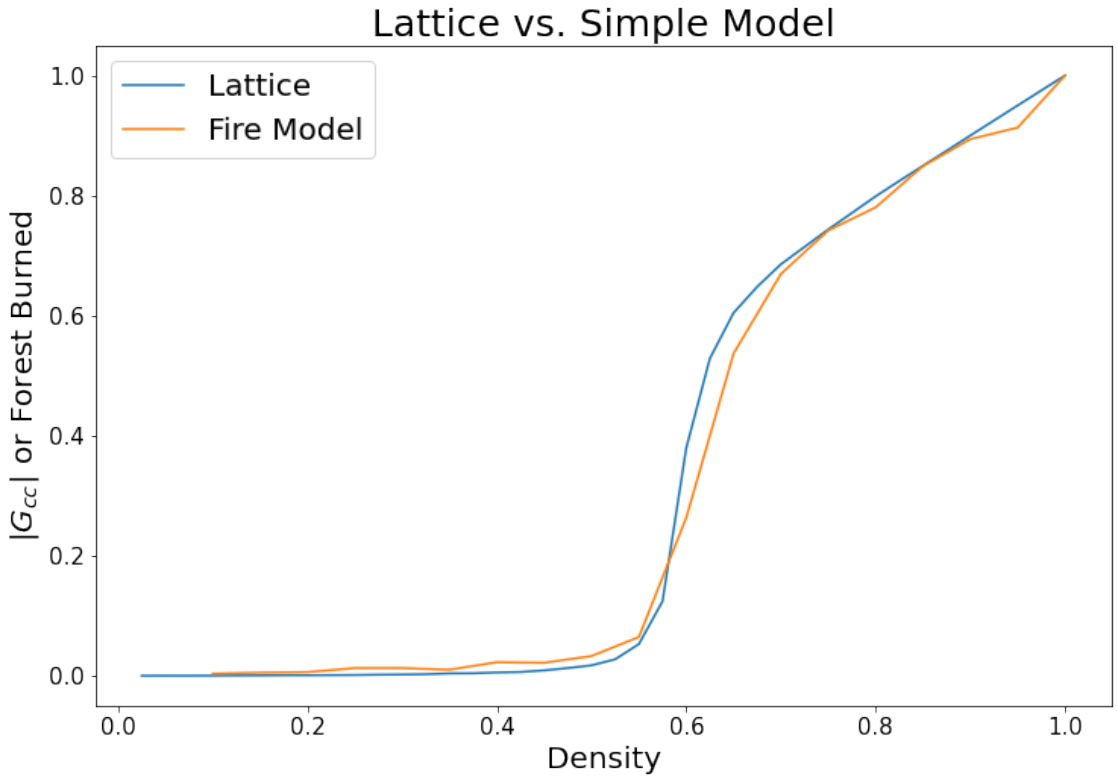

Figure 4.7: Nonlinear relationship between density and total burn

It is immediately apparent from these results that the relationship between a deterministic wildfire model and the largest connected component in a lattice is seemingly identical. While the characteristics of the largest connected component in a trimmed lattice have yet to be formally proved, the author believes it shows the powerful motivation for applying network theory to better understand the behavior of potential wildfires on an environment. It is of the authors final opinion that a better understanding of the properties of an environments network structure could yield many insights into the behavior of fires on any given environment.

Chapter 5. Conclusions

5.1 Summary

While modeling wildfire is a difficult problem to solve, stochastic agent-based models are a promising method to achieve viable results. We have shown that the proposed model produces realistic behaviors one would expect in a real wildfire, and produced a flexible framework to perform uncertainty quantification and sensitivity analysis on any given wildfire model parameters. Lastly, we illustrated our models close relationship with network theory, and how it applies to optimal control principles using the max-flow min-cut theorem, as well as proposed some possible connections regarding the correlation of largest connected components of a randomly trimmed lattice graph and the total burn area of a deterministic agent-based wildfire model.

5.2 Future Work

Given more time to continue this research, there are two main areas upon which research would continue. The first, would be to devise a method to build environments from satellite images of known fires, and test the accuracy of the agent-based model. The purpose of this would be to leave the idealized environments of the grasslands, mountains, and golf course, and focus on model output looks like in real world scenarios. Additionally, these known fires and respective fire boundaries would lead to better parameter estimation in the agent-based model, leading to improved model performance. The second area of improvement would be additional research in representing environments as networks, and applying many of the tools to simplify or explain the future behavior of a fire based off of its network structure. Both these areas have interesting implications of feasibility for agent-based models to correctly predict fire behavior, as well as discovering novel ways to build understanding of fire spread through the properties of its underlying network.

A Accessing The Code

All code is kept in a public repository on Github, available at: `https://github.com/Ging3y/BYUMathFire_DM/` There is both a dedicated `README`, as well as example code to create environments, load environments, alter conditions through configuration files, add agents, produce visuals, and create and save gifs of fire spread. For development, the *Spyder* IDE is recommended due to its plot window that allows for easy testing and visualization of the environment evolution.

When running longer simulations, it is advisable to disable plotting in order to speed up calculations, and simply save results at the end of the simulation. For a 100x100 environment forecasting 100 iterations into the future, computation time on a single core is an average time of 9.5 seconds using a 2.6 GHz 6-Core Intel Core i7. The code is written in such a way that it is embarrassingly parallel, being able to run various simulations on multiple cores. Unfortunately, memory restrictions grow quite quickly, with the approximate total bytes required per simulation to be $256 \times l \times w$, due the necessary storage of the fuel, topology, state, and agent maps for each cell in the array at 64 bit precision.

Bibliography

[1] Wallace, L. "Analysis of fire spread in light forest fuels," Journal of Agricultural Research, vol. 72, no. 3, p. 93, 1946.

[2] Emmons, H. "Fundamental problems of the free burning fire," in Symposium (International) on Combustion, vol. 10, no. 1. Elsevier, 1965, pp. 951–964.

[3] Claire Miller, James Hilton, Andrew Sullivan, Mahesh Prakash. SPARK – A Bushfire Spread Prediction Tool. 11th International Symposium on Environmental Software Systems (ISESS), Mar 2015, Melbourne, Australia. pp.262-271, ff10.1007/978-3-319-15994-2_26ff.

[4] Finney, M. A. (1998). FARSITE: Fire Area Simulator - Model Development and Evaluation. United States Department of Agriculture - Rocky Mountain Research Station.

[5] Tolhurst, Kevin Shields, Brett Chong, Derek. (2008). Phoenix: Development and application of a bushfire risk management tool. Australian Journal of Emergency Management. 23. 47-54.

[6] Congressional Research Service. (2021, January). Wildfire Statistics. https://fas.org/sgp/crs/misc/IF10244.pdf

[7] National Fire Protection Association. (2020, November). Greetings from the 2020 Wildfire Season. https://nfpa.org/News-and-Research/Publications-and-media/NFPA-Journal/2020/November-December-2020/Features/Wildfire

[8] Bureau of Land Management. About: What we manage: National: Bureau of Land Management. National — Bureau of Land Management. Retrieved December 7, 2021, from https://www.blm.gov/about/what-we-manage/national.

[9] Huang, Q., Razi, A., Afghah, F., & Fule, P. (2020). Wildfire spread modeling with aerial image processing. 2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM). https://doi.org/10.1109/wowmom49955.2020.00063

[10] Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating

human systems. Proceedings of the National Academy of Sciences of the United States of America, 99.

[11] Cilfone, N. A., Kirschner, D. E., Linderman, J. J. (2014). Strategies for efficient numerical implementation of hybrid multi-scale agent-based models to describe biological systems. Cellular and Molecular Bioengineering, 8(1), 119–136. https://doi.org/10.1007/s12195-014-0363-6

[12] Steven C. Bankes, Proceedings of the National Academy of Sciences May 2002, 99 (suppl 3) 7199-7200; DOI: 10.1073/pnas.072081299 Agent-based modeling: A revolution?

[13] Rothermel, R. C. (1972). A mathematical model for predicting fire spread in wildland fuels. USDA Forest Service. https://www.fs.fed.us/rm/pubs_int/int_rp115.pdf

[14] Weise, D. R., Cobian-Iñiguez, J., Princevac, M. (2018). Surface to crown transition. Retrieved from https://www.fs.fed.us/psw/publications/weise/psw_2018_weise003.pdf

[15] Mandel, J., Bennethum, L. S., Beezley, J. D., Coen, J. L., Douglas, C. C., Kim, M., Vodacek, A. (2006, June). A wildland fire model with data assimilation. arXiv. Retrieved from https://arxiv.org/abs/0709.0086

[16] Wilensky, U. (1997). NetLogo Fire model. http://ccl.northwestern.edu/netlogo/models/Fire. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

[17] Saltelli, A.; Ratto, M.; Andres, T.; Campolongo, F.;Cariboni, J.; Gatelli, D.; Saisana, M.; Tarantola, S. *Global Sensitivity Analysis: The Primer;* John Wiley & Sons: Hoboken, JN, USA, 2008.

[18] Finbow, S. (2009). The Firefighter Problem: A survey of results, directions and questions. Australasian Journal of Combinatorics, 43, 57–77.

[19] Newman, M. E. J. (2019). Random Graphs. In Networks. essay, Oxford University Press.

[20] Cotter, S. C. (1979). A screening design for factorial experiments with interactions. Biometrika, 66(2), 317–320. https://doi.org/10.1093/biomet/66.2.317

[21] Robertson, D., Zañartu, M., & Cook, D. (2016). Comprehensive, Population-Based Sensitivity Analysis of a Two-Mass Vocal Fold Model. PloS one, 11(2), e0148309. https://doi.org/10.1371/journal.pone.0148309