Forest Fire Analytics: A Comprehensive Network Model Study

Gian Alix*
gian.alix@gmail.com
login: royalle
214760870

Baoqi Yu*
baoqiyu@my.yorku.ca
login: baoqiyu
215118425

Jing Li*
jliellen@my.yorku.ca
login: jliellen
216026999

KEYWORDS

fire propagation, network

1 INTRODUCTION

Efficient forest fire control is one of the most challenging and important problems. Wildfires have resulted in irreversible environmental and socio-economic damages across the world. According to the National Interagency Fire Center, as of November 2020, wildfires have burned more than 8 million acres in California [6]. A better understanding of fire propagation is needed to facilitate fire control and mitigate those damages. A number of models have been developed to simulate wildfire propagation in previous studies. In particular, networks have been extensively applied. In this project, we focus on simulating fire propagation through integrating topographical and weather conditions in the model.

The remainder of this report is organized as follows: **Section 2** defines the problem. The related work is presented in **Section 3**. **Section 4** presents the methodology. After showing the experiment results in **Section 5**, we conclude in **Section 6**.

2 PROBLEM DEFINITION

The focus of this project is on simulating fire propagation in lattice networks (or grid networks). In particular, we aim to address the following problems:

- Problem 1: Given an area, construct a lattice network G, and transform elevation, slope, aspect and wind conditions into the network through linear threshold model.
- Problem 2: Given a lattice network, simulate wildfire propagation in the network.
- Problem 3: Given a lattice network, model the effect of landmarks such as roadways on wildfire propagation.

3 RELATED WORK

3.1 Forest Network

Hajian et al. [5] obtain the Voronoi-based network by dividing a map into many homogeneous sub regions. In their study, sub regions are derived by overlapping different fire environmental data layers through Geographic Information System (GIS). An inside point is assigned to represent each sub region and Delaunay triangulation is then employed to construct the network edges. The constructed network owns significant accuracy and reality. However, most of these data layers have not been made public, and the availability of GIS applications is limited.

Based on the Cellular Automata (CA), another type of network used in fire propagation modelling is lattice graph [7, 20]. It utilizes regularly spaced sample points to represent landscapes. Fire propagates through the grid-cell basis. The advantage of this network is that weather conditions as well as land topography can easily be incorporated into the model [7, 20]. Pais et al. [14] partition off the forest landscape into a series of identical area square cells. In their study, topographic data and weather conditions are integrated in the model to simulate fire growth.

3.2 Fire Propagation

Topography environment (e.g., elevation, slope, aspect) influences fire behavior quantitatively[1, 4, 9–11, 16]. In particular, the steeper the slope the faster the fire will spread [4, 9]. The aspect determines the amounts of solar radiation, moisture, and wind a slope receives. This could lead to varied temperatures and drier conditions, which can also contribute to fire behavior directly through different composition in vegetation and density [1, 4, 11]. On the other researches, discussed burn severity was negatively correlated with elevation [10, 16].

Nelson et al. [12] analyze fire propagation and wind speed. Their experiment suggests the fire spread rate is proportional to the square of the wind speed. There are three types of fire, namely: surface fire, crown fire and underground fire. Under strong wind conditions, the fire will burn to the crown and the fire spread rate will increase [12]. Werth et al. [19] studies fire propagation and wind direction. They point out the spread of fire is nearly round under low wind, while under strong wind it is elliptic, and its long axis is parallel to the direction of the leading wind.

3.3 Linear threshold model

The Linear Threshold (LT) model has been widely used in modelling diffusion process. Pathak et al. [15] present a generalized version of the linear threshold model for simulating multiple cascades on a network while allowing nodes to switch between them. It also has been used in social networks. Chen et al. [3] proposed a scalable algorithm to find a small set of most influential nodes under the linear threshold model. In our work, we will employ this model in the network to simulate fire propagation.

4 METHODOLOGY

In order to address the problems as discussed in **Section 2**, we take the following steps:

(1) Secure the dataset, which is discussed briefly in **Section 4.1**.
(2) Leveraging this data, the network will then be constructed. This process will be thoroughly covered in **Section 4.2**.
(3) After the network has been constructed, we can finally perform some simulations and run some tests. **Section 5** should be able to report the detailed analyses and results of the performed experiments.

*All authors contributed equally to this research.

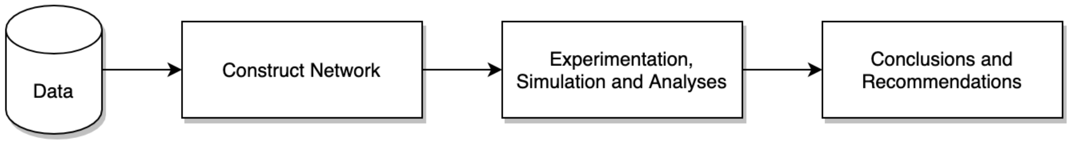

Figure 1: The high-level framework to attempt to solve the problem as discussed in Section ?? involves four major steps: (1) Acquisition and Analyses of Data, (2) Network Construction, (3) Experimentation, and (4) Conclusion and Recommendation.

(4) Gaining some insights, we finally give some concluding remarks & recommendations on how the problem can be extended.

This high-level framework of our action plan can be seen visually in **Figure 1**.

4.1 Dataset

The dataset that we use to construct the network is the **ForestCoverType** [2]. This dataset contains tree observations (30 meter × 30 meter grids of forests) of the Roosevelt National Forest in Colorado. Each grid includes several topographic data such as elevations, slopes and aspects.

4.2 Network Constructing

To model wildfire propagation in heterogeneous forest landscapes, the terrain of the Roosevelt Forest in Colorado is tessellated into a number of small patches whose shape are squares as seen in **Figure 2** [18]. As such, the terrain is transformed into a lattice network $G(N, E)$ of dimension $|N| \times |N|$, where $N = \{v_k\}$, with $k = 1, 2, ..., |N|$ is the set of nodes (the terrain patches or the forest nodes), and E is the set of edges (links) between neighbor nodes. In the context of forest fire propagation, these links will serve as travelling routes for fire to spread from one forest to a neighboring one.

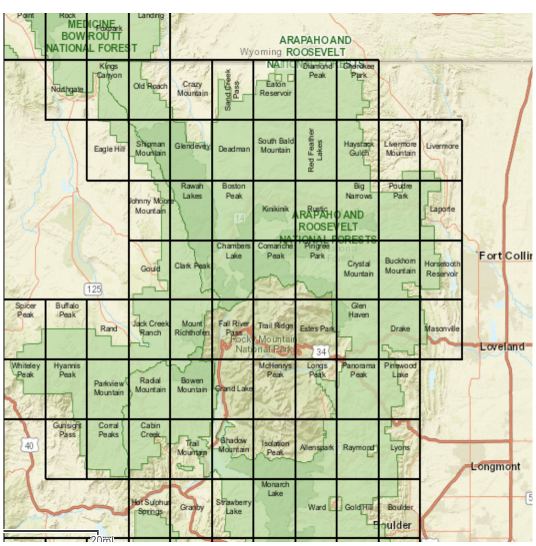

Figure 2: The above picture displays the grid map of the Roosevelt National Forest.

Each node in the network graph has a state that serves as one of its node attributes. At a time point t, a node state can be in any one of the following states: *empty*, *not_burned*, *burning*, or *burnt*. The "empty" represents fire zones or areas where there are no trees (cannot be burned). The state "not_burned" represents trees are intact. The states "burning" and "burnt" represent trees currently burning and have been burnt down respectively. A user-defined density factor ρ is then used to determine the forest density of our network, where links between adjacently-placed neighboring forests is determined by this probability factor.

The fire propagation probabilities are computed using a linear threshold model. Each node v_k uses three topographic features, namely the elevation, slope and aspect, as factors to compute the node's threshold θ. This indicates how easily v_k would switch from one state to another. Each edge (v_k, v_l) is weighted and the weights can be calculated using wind and Euclidean distance factors of v_k and v_l. More details on the linear threshold model, along with the mathematical models, shall be explained in the next sections. Thus, the spatial distribution of fire propagates is converted to a cascading problem in networks.

4.3 The Modified Linear Threshold Model

A modified version of the LT model as discussed in **Section 3.3** was employed in the simulation of the forest fire propagation. In this probabilistic diffusion-based network model, nodes can have two possible states: *active* and *inactive*. Therefore, in the context of our forest network, active nodes are those forests that are currently burning (i.e. having 'burning' state) while inactive ones are those that have not yet been burnt (i.e. having 'not_burning' state). Clearly, those nodes in the system that are quoted as 'empty' have no role to play in this LT model. And those burnt out forests can be thought of as 'inactive' in the LT model as fire has already been extinguished and can no longer influence neighboring forests. However, these burnt nodes while considered inactive loses their role to play in the LT model as burnt out forests cannot go back to being in the burning state.

Each node then in the network will have some uniformly random probability $\theta \in [0, 1]$ which serves as a threshold for when its state flips. One tweak proposed by our team is that we can change the way we pick these uniformly selected random numbers. A mathematical model to compute these node thresholds is described in full detail in **Section 4.4**. Then each edge in the (undirected) network will have some randomly assigned weight $\beta \in [0, 1]$. Similar to node

Table 1: Model Parameters used for the Node Threshold θ and the Edge Weight β

Symbol	Nomenclature	Formula	Notes
C_n	node stochastic component	random$[0, 1]$	-
ϕ_s	slope coefficient	$5.275(\tan\phi)^2$	Rothermel [17]; ϕ is the slope in the ForestCoverType dataset [2]
ξ	elevation coefficient	$\frac{1}{1+\ln(\max\{he^{-6}, 1\})}$	Olabarria, et. al. [13]; h is the elevation in the ForestCoverType dataset [2]
α	aspect coefficient	(See **Table 2**)	Estes, et. al. [4]
C_e	edge stochastic component	random$[0, 1]$	-
ϕ_w	wind speed	random$[0, \psi]$	ψ is the max speed as defined by the user
τ	wind direction	$\arctan\left(\frac{\Delta y}{\Delta x}\right)$	Calculated: Δ denotes the horizontal (x) and vertical (y) node distances
δ	node Euclidean distance	$\sqrt{\Delta^2 x + \Delta^2 y}$	Calculated: Δ denotes the horizontal (x) and vertical (y) node distances

thresholds, we detail in **Section 4.5** the mathematical formula in the calculation of edge weights. Thus, for any inactive node (or not burning forest) v in the forest network, its state will switch to an active (or burning) state if and only if the total weights of all edges of (u, v) is greater than the node threshold of forest v, where u is an active (burning) neighbor of v. [8] Mathematically speaking:

$$\left(\sum_{u\in[\text{active neighbor of } v]} w_{u,v} > \theta_v\right) \Longrightarrow \text{switch state of } v \quad (1)$$

The process initially begins with all nodes being inactive. A random node is then selected to be active and the diffusion process happens for that node (cascading to its neighbors). To translate this in the context of our forest network system, we begin with an initial setup where no forests are burning. Then a forest is selected at random, which serves as the ignition source of the forest fires. Forest fire propagation should then begin to take into effect and will cascade to its neighbors and to its neighbors' neighbors, according to the mathematical process as described by **Eq. (1)**. This cascading process is repeated for all nodes in the network, and the following are the stopping criteria for when this iterative process halts:

- If we have reached the maximum number of timestep iterations, which can be parameterized
- When all nodes have been switched from inactive to active, or in this case if all non-burning forests have turned burning or have become burnt
- When no new node or forest has become active or have been switched to the burning state

The algorithms as discussed in **Section 4.6** should give a better understanding of how the linear threshold model can be implemented in the simulation of forest fire propagation.

The tweak that the team proposed to modify the the method for selecting node thresholds and edge weights is rooted from the limitation of the basic LT model. In the basic model, node thresholds and edge weights are selected uniformly at random. Here, the rationale for this choice of method of picking random values was the lack of knowledge of the network, as identified by Kempe et al. [8]. To some extent, the randomness of these probability values check out due to the stochasticity of fire spread. Clearly, the LT model can be utilized to model fire propagation but being able to take into account the various fire spread factors involves designing new mathematical models that can compute the node threshold and edge weight values. These innovative models should also be designed to preserve the model's cascading effect.

4.4 The Mathematical Formulation of the Node Thresholds

In this section, we attempt to discuss thoroughly the mathematical model behind the formulation of the nodes' thresholds. Each node in the network graph is assigned node threshold θ, which can be calculated as:

$$\theta = \frac{1}{\pi}\arctan(C_n\phi_s\xi\alpha) + 0.5 \quad (2)$$

While the node thresholds may appear to be as random as the basic Linear Threshold model due to the stochastic component C_n, the various factors that affect the cause of fire (such as slope, elevation and aspect reflected by the parameters ϕ_s, ξ, α respectively) are also taken into account in the formula. **Table 1** presents us with a tabulated summary of the node threshold symbol, nomenclature, and formula for such parameters. For instance, the slope coefficient ϕ_s can be obtained using the formula

$$\phi_s = 5.275(\tan\phi)^2 \quad (3)$$

where ϕ is the slope found in the **ForestCoverType** dataset [2]. The direct relationship between slope steepness and likelihood of fire is dictated in this formula and that this is actually from a portion of Rothermel's fire spread model [17]. In fact, many pieces of various existing fire propagation math models were used and incorporated in our mathematical models. Another example can be seen for the formula for calculating the elevation coefficient ξ, which was borrowed from Olabaria, et. al. [13] and can be calculated by:

$$\xi = \frac{1}{1+\ln(\max\{he^{-6}, 1\})} \quad (4)$$

where h is the elevation value and again can be found in the **ForestCoverType** dataset [2]. It can be noted that Olabaria, et. al. has formulated this mathematical expression based on their dataset, with elevation values $h \in [0, 2300]$ [13] while the **ForestCoverType** dataset uses a different range of values for the elevation, with $h \in [1859, 3858]$ [2]. That said, we can map the values of h that we have in our dataset and transform them into elevation values used within Olabaria et al.'s elevation range [13]. A simple min-max

Table 2: A Lookup Table for the Aspect Coefficient α. Note that the aspect value A can be found under the 'Aspect' attribute in the ForestCoverType dataset [2].

Aspect Value A	Direction	Aspect Coefficient α
$A \in [0, 22.5) \cup [337.5, 360]$	North	-0.063
$A \in [22.5, 67.5)$	North-East	0.349
$A \in [67.5, 112.5)$	East	0.686
$A \in [112.5, 157.5)$	South-East	0.557
$A \in [157.5, 202.5)$	South	0.039
$A \in [202.5, 247.5)$	South-West	-0.155
$A \in [247.5, 292.5)$	West	-0.0252
$A \in [292.5, 337.5)$	North-West	-0.171

normalization technique can be employed for the purposes of this simple transformation in order to utilize their formula. Additionally, it can be noticed that the elevation and probability for fire are negatively correlated variables. This means that the higher the elevation values of a forest, the lower its risk for it to catch on fire. The rationale behind such claim is that higher elevation lands tend to receive less forest fuels than those in lower-elevated places. As a result, higher places are less prone to catch fires than those in the lower areas. Hence, this should clearly explain why elevation presents itself to be an inverse factor in the mathematical model, which was also observed by Olabaria, et. al [13].

We then turn our attention to the aspect coefficient α. The study by Estes et al. [4] provides us with information on how we can obtain the aspect coefficient α based on the aspect value A . A lookup table can be found on **Table 2**. Again the aspect values can be found in the **ForestCoverType** dataset [2], whose values range from $A \in [0, 360]$.

In the math formula for the node threshold found in **Eq. (2)**, we can see that the function applied is $\arctan(\cdot)$ and the choice of such function is arbitrary. In fact, any function would have worked so as long as the function is monotonous. The range of the values of $\arctan(\cdot) \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, which should clearly explain the purpose of $\frac{1}{\pi}$ that is merely used as a scaling factor. Hence this particular constant can change depending on the choice of function used in place of $\arctan(\cdot)$. Furthermore, the additional 0.5 constant should give an idea that it is a correction factor as the first term has range of values from $[-0.5, 0.5]$. Hence, our mathematical model for node threshold should be some valid probabilistic number between $[0, 1]$.

While the mathematical formula is not as perfect as it is compared to existing models, it is a good place to start. In fact, it may be that this particular model would have to be redesigned in our next steps in order to reduce the contribution of the stochasticity component.

4.5 The Mathematical Formulation of Edge Weights

Like the mathematical model used for modelling the node thresholds, we also have a mathematical model used to compute the weights of edges in our network graph. The weight β of the edges can be computed as follows:

$$\beta = \arctan\left(\frac{C_e \phi_w \cos\tau}{\delta}\right) \tag{5}$$

where the symbol, nomenclature and formulas can be found in **Table 1**, similar to where the node threshold parameters are tabulated. Similar to the modelling of node thresholds, the modelling of edge weights also relies on a stochastic component $C_e \in [0, 1]$ chosen uniformly at random and is independent from the C_n choices for the two nodes $\{n_i, n_j\}$ connected by the edge (n_i, n_j). Again, the model is not perfect and a more robust math formula may be designed as a next step. The parameter ϕ_w is the wind speed and is a random number between $[0, \psi]$ where ψ is a user-defined parameter that determines the maximum possible wind speed at any given time t. The factor $\cos\tau$ gives contribution to the angle direction of the wind speed, where the angle τ can easily be computed based on the positions of the nodes in the network graph. And finally the Euclidean distance δ, also based from the positions of the nodes in the network, is a another factor considered that should reduce the influence of fire propagation spread for whenever there is a large distance between any two forests.

4.6 The Algorithms

In this section, we briefly discuss the algorithms employed in solving the problem at hand. We first turn our attention to **Algorithm 1** and **Algorithm 2**. It can be observed that the IGNITE algorithm is simply a sub-routine of the main SIMULATE-FIRE method. It can also be observed that SIMULATE-FIRE is simply the procedure that was discussed in **Section 4.3**. In Lines 1-2, some random forest in N is selected at random and this will serve as the ignition source of fire in the network. In Line 3, we introduce three new variables that represent the set of currently burning forests B, the set of burning forests in the previous timestep $\hat{B}$, and the iterator k. The cascading process then comes into play in Lines 4-7 where the stopping criterion is when either the number of iterations has reached the maximum (parameterized by t), or when the set of currently burning forests is the same set of burning forests in the previous timestep (i.e. no new forests have switched to the burning state). Exploring the `while` loop, in Line 5 we perform the IGNITE procedure of the graph G (see below for details). Then in Line 6, we let the current set of burning forests to be the set of forests burning in the previous timestep. And then obtain the new set of currently burning forests. Then increment our iterator in Line 7 to prepare for the next iteration of the loop.

We then have a look at the IGNITE sub-algorithm. In Line 1, Γ will contain all those forest nodes that are burning. We then operate a `for-each` loop in Lines 2-12, where for each burning node v in Γ, we do Lines 3-12. We gather all the neighbors of forest v and store them into Λ. In Lines 4-5, we check if the burning forest v does not have neighbors; in that case, we simply break off the loop. Otherwise, in Lines 6-12, we take each of those neighbor nodes of v that was stored in Λ. And then, for each n in Λ, we obtain all neighbors of n that are burning and store them in Ψ as seen in Line 7. Lines 8-10 computes for the total sum of the weights of the adjacent edges of n and caps it off at 1, to keep the probability values within $[0, 1]$. In Line 11-12, we check if the sum of those edges surpass the node threshold of node n. If this is the case, then

Algorithm 1: IGNITE $(G(N, E))$

Input : The forest graph G, with forest set N and probable fire path set E

Output : (None). The resultant forest graph of the cascading Linear Threshold model

$\Gamma \leftarrow \{f \mid f \in N \wedge f.\text{fire\_state} = \text{'burning'}\}$
foreach v *in* Γ **do**
    $\Lambda \leftarrow v.\text{neighbors}$
    if $\Lambda = \emptyset$ **then**
        break
    foreach n *in* Λ **do**
        $\Psi \leftarrow \{t \mid t \in n.\text{neighbors} \wedge t.\text{fire\_state} = \text{'burning'}\}$
        $s \leftarrow 0$
        foreach u *in* Ψ **do**
            $s = \min(1, s + \beta_{u,v})$
        if $s > \theta_n$ **then**
            $n.\text{fire\_state} \leftarrow \text{'burning'}$

Algorithm 2: SIMULATE-FIRE $(G(N, E), t)$

Input : Graph G with set of nodes N and set of edges E

Output : (None). The resultant G with wind simulated

$u \leftarrow$ pick a random forest $\in N$
$u.\text{fire\_state} \leftarrow \text{'burning'}$
$B \leftarrow \{u\}; \hat{B} \leftarrow \emptyset; k \leftarrow 1$
while $k \leq t \wedge B \neq \hat{B}$ **do**
    IGNITE$(G(N, E))$
    $\hat{B} \leftarrow B; B \leftarrow \{f \mid f \in N \wedge f.\text{fire\_state} = \text{'burning'}\}$
    $k \leftarrow k + 1$

Algorithm 3: SIMULATE-WIND (G, ψ)

Input : Graph G, and a maximum possible speed ψ

Output : (None). The resultant G with wind simulated

1 $\gamma \leftarrow \text{random}[0, \psi]$
2 $n \leftarrow$ pick a random forest, with $pos = (n_x, n_y)$
3 $r_1, r_2 \leftarrow \{a, b \mid a, b \in \text{random}(\mathbb{Z}^+)\}$
4 $\Omega \leftarrow [n_x - r_1, n_x + r_1] \times [n_y - r_2, n_y + r_2]$
5 $E_{\text{update}} \leftarrow \{(n_i, n_j) \mid n_i, n_j \in \Omega\}$
6 $\Delta(G, E_{\text{update}}, \text{'wind'}, \gamma)$

the forest node n switches to the 'burning' state. This is what is happening under the hood of IGNITE, which should give a clearer picture of the main method SIMULATE-FIRE.

Not only did our team simulate fire but also the wind (it shall be discussed in later sections that wind has not yet been incorporated to the modelling for edge weights due to some constraints). However, we shall discuss the algorithm of SIMULATE-WIND as seen in **Algorithm 3**. We take a graph G as input, along with the user-defined parameter ψ, which should act as the maximum possible wind speed. In Line 1, we pick that wind speed $\gamma \in [0, \psi]$ which will be picked uniformly at random. In Line 2, we pick a forest n in the network at random and we suppose that it has position (n_x, n_y). We then pick in Line 3 a pair of positive integers at random (r_1, r_2). Ideally, we want to pick values within the range of our network lattice. However, we have uplifted this restriction by allowing out-of-bounds results while getting the wind simulation to still working properly. By Line 4, we determine the Area of Effect (AoE). This is simply an elliptical region for where all edges lying in this particular region will experience changes to the wind conditions. At closer inspection the region bounded by Ω seems to be rectangular; however what this area presents is actually elliptical in shape and that only the ranges of the major and minor axes of the ellipse are presented. In Line 5, we store all those edges within the elliptical boundary into the set E_{update}. And then in Line 6 apply those changes to the Graph G, i.e. update the 'wind' attribute of the edges in E_{update} from G with the randomly selected γ value.

A sample simulation that employs these algorithms can be seen in **Figure 3** and **Figure 6**. A more thorough discussion of these simulations can be seen in **Section 5**.

5 EXPERIMENTS / EVALUATION

5.1 The Network

The graph has 100 nodes in the experiments, but the size is scalable. Each node has a chance to have an edge to its adjacent nodes with probability ρ, the network's density factor. The standard base density factor chosen for all experiemnts is $\rho = 0.8$.

5.2 Experiments

All experiments were conducted on a MacBook Pro with Dual-Core Intel Core i5 CPU @ 2.3 GHz and 8GB memory, using Python 3.7.4. All models are generated by the `NetworkX` package. Experiments and results are tested and validated using a MacBook Air with Dual-Core Intel Core i5 CPU @ 1.6 GHz and 8GB memory.

We have tested and run the forest fire propagation algorithm in our forest network system around 100 times. A sample sequence of the changes to the forest system can be observed in **Figure 3**.

5.3 Analyses

We have performed some experiments and simulations. Firstly, we looked into the percentage of forests in the map that are both burning and burnt down overtime. A plot of this can be seen in **Figure 4** and our team observed some interesting results. For one, which may seem intuitive is that the percentage of damaged forests (which is simply the total number of burning and burnt forests) tends to converge to ρ, the density factor of the system. This suggests that all (or perhaps almost all) of the forests will eventually be impacted by the fire, especially for higher values of ρ that implies a more connected forest network. Details on the analysis of ρ will be discussed later. Another thing to notice is that the number of burning forests would initially be greater than the number of burnt forests in the system until at some point before the number of burnt forests will be larger. This we saw was always true and that there seems to exist a time step threshold t' such that past this point, then the number of burnt forests is greater than the number of of burning forests. Another clear observation is the shapes of the plots for the number of burnt and damaged forests. They seem to follow the same shape patten, with the exception that the number of burnt

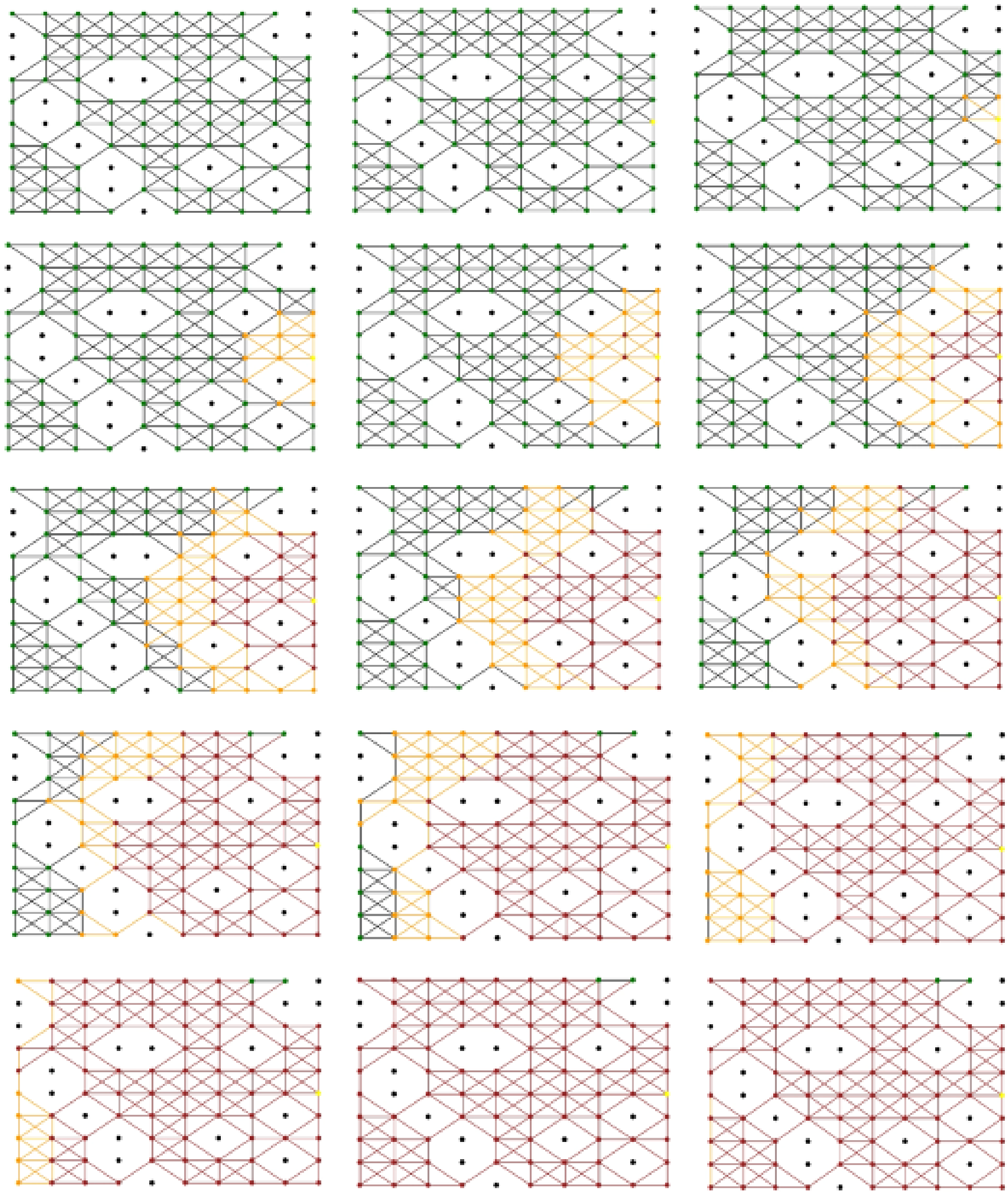

Figure 3: This is a sample run of the forest fire propagation simulation (sequence begins in the upper left corner going across, then down). The figure in the upper left corner panel is the initial state of the system where none of the forests are damaged (all non-burnt forests are green and all those fire zones whose states will never change are black). In the figure beside it shows a randomly selected node in the forest (colored yellow) where it serves as the source of the fire. In the sequences that follow, we see that fire propagates accordingly to our algorithms (where orange nodes represent forests that are burning and brown nodes represent forests that are burnt out). In the final few panels of the simulation, we have seen that the number of burning forests has remained the same and therefore serves as the stopping criterion of the fire propagating algorithm.

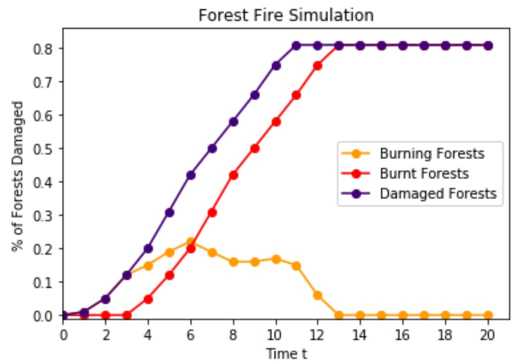

Figure 4: The above shows a plot overtime comparing the percentage of the area damaged: the fraction of those forests burning, the fraction of those that are burnt down, and the fraction of those that have been damaged (note that the total damage is simply the number of forests that are burning and the number of forests that have been burnt.

forests is simply shifted to the right. And lastly, the percentage of the burning forests will always have this shape: that is, it will increase until it hits some threshold (say λ) and then past this λ, it will either decline or it will plateau then decline. Afterwhich, it will stagnate to 0 until the end of time. These are clear observations and pretty much intuitive. Fire will spread out, propagate and then just die out.

More or less, the observations and analyses conducted from **Figure 4** are pretty much the same (or almost the same) for almost all runs and experiments we have tested on. However, we did notice changes for when we attempted to crank up the density factor ρ. In **Figure 5**, we have attempted using various ρ values. Aside from $\rho = 0.8$, we also used $\rho = 1.0$ and $\rho = 0.5$. For the situation for when $\rho = 1.0$, this suggests that the entire system is densely forested and that all nodes/forests are connected together. As a result, fire propagates at a much faster rate and it can be observed from the

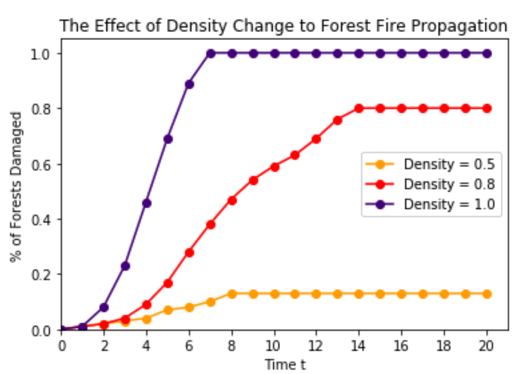

Figure 5: The above shows a plot overtime of the percentage of forests damaged for different values of the forest density factor ρ. This includes both those forests that are burning and those that are burnt.

plot that as early as time $t = 6$, the entire forest system has already burnt down. And we can see that this is not the case for the other ρ values. When $\rho = 0.8$, it takes 14 time steps before the entire forest system burns down. However, it is a different case for when $\rho = 0.5$. In this setting, roughly only half of the system is forested. This implies that some forests may be disconnected from others due to a great number of fire zones or areas that are not burnable. As a result, forest fires do not propagate easily to all areas and from the plot that we see in **Figure 5** that roughly only 10% of those forested areas have been damaged. It does not also support the claim that the percentage of damaged areas approach ρ in the long run as this is not the case for when $\rho = 0.5$. Perhaps this is the case for all scenarios for when $\rho > \rho'$ but not so for when $\rho \leq \rho'$ (such as when $\rho = 0.5$). Therefore, the plot as displayed in **Figure 5** verifies our hypothesis that the more connected and the more dense the forest network is, the more likely it is for fire to spread. Additionally, fire propagates at a much faster rate and more forests will thus get destroyed at an earlier time point. Furthermore, this implies that the density factor ρ indeed does play a big role in the simulation of spread of forest fires.

We have also explored on the simulation of wind to the forest network, in accordance with **Algorithm 3**. An example of this simulation that runs over 3 time steps can be seen in **Figure 6**. In

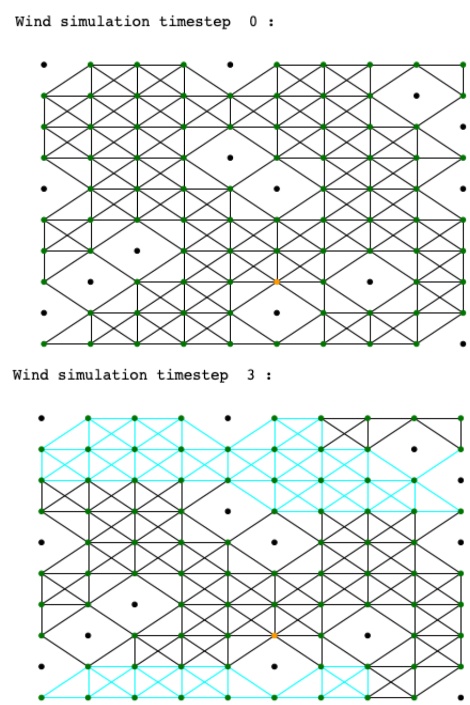

Figure 6: Wind simulation in the Forest Network where wind conditions change at every $t = 1$ time steps (only the initial and final lattice network is shown after 3 timesteps). The edges in cyan color indicate that there were changes with the wind condition at any given time from the initial time of simulation.

the simulation for some t timestep, the algorithm chooses a random forest in the network which acts as the center or the "eye" of the area affected by the wind, and nearby edges of the node in the network are considered affected by the change of wind conditions. As one could imagine, this is similar to how wind behaves in a storm or typhoon.

6 CONCLUSIONS

As recap of the work completed, we have constructed a lattice network to simulate forest fire propagation and have also incorporated topographic features such as elevation, slope and aspect in networks under the modified Linear Threshold model. Forest fire propagation has been simulated several times under the algorithm described by the LT model, using probabilities that can be computed using the mathematical models as proposed by the team. Various testing and experiments have been conducted and insightful results have been reported. One remarkable result that validates our intuitive hypothesis is that the the forest network density factor ρ plays a significant role in the proprgation of forest fires.

Unfortunately, we have not yet incorporated wind into the computation of edge weights. This limitation is due to the fact that our team may still need an in-depth review of the model for the edge weights and that further testing may still be required. Upon careful consideration, it was decided that wind factors are left out at this stage of our project. However, we still plan on incorporating this component as wind is an important variable that plays a critical role in the simulation of forest fires. We expect the wind phenomenon to materialize in our next steps. For now, we were able to simulate wind (without fire) as a stepping stone to achieve our desired goal and this ability to do so should still prove to be significant.

For our next steps, in addition to the wind factor, we identify three extensions to our problem: (1) determine how landmarks such as roadways and villages can affect wildfire propagation, (2) model a more realistic setup such as long-range spotting fires, multiple ignition sources, and forest growth (i.e. burnt out forests grow trees that can still change state to 'burning'), and (3) propose further intervention strategies to reduce the spread of wildfires.

REFERENCES

[1] Patricia M Alexandre, Susan I Stewart, Miranda H Mockrin, Nicholas S Keuler, Alexandra D Syphard, Avi Bar-Massada, Murray K Clayton, and Volker C Radeloff. 2016. The relative impacts of vegetation, topography and spatial arrangement on building loss to wildfires in case studies of California and Colorado. *Landscape Ecology* 31, 2 (2016), 415–430.

[2] Jock Blackard, Denis Dean, and Charles Anderson. 1998. Covertype Data Set. https://archive.ics.uci.edu/ml/datasets/covertype

[3] W. Chen, Y. Yuan, and L. Zhang. 2010. Scalable Influence Maximization in Social Networks under the Linear Threshold Model. In *2010 IEEE International Conference on Data Mining*. 88–97. https://doi.org/10.1109/ICDM.2010.118

[4] Becky Estes, Eric Knapp, Carl Skinner, Jay Miller, and Haiganoush Preisler. 2017. Factors influencing fire severity under moderate burning conditions in the Klamath Mountains, northern California, USA. *Ecosphere* 8 (05 2017). https://doi.org/10.1002/ecs2.1794

[5] Mohammad Hajian, Emanuel Melachrinoudis, and Peter Kubat. 2016. Modeling wildfire propagation with the stochastic shortest path: A fast simulation approach. *Environmental Modelling Software* 82 (2016), 73 – 88. https://doi.org/10.1016/j.envsoft.2016.03.012

[6] K. Hoover and L. A. Hanson. 2020. Wildfire Statistics. (2020). https://fas.org/sgp/crs/misc/IF10244.pdf

[7] Ioannis Karafyllidis and Adonios Thanailakis. 1997. A model for predicting forest fire spreading using cellular automata. *Ecological Modelling* 99, 1 (1997), 87 – 97. https://doi.org/10.1016/S0304-3800(96)01942-4

[8] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the Spread of Influence through a Social Network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Washington, D.C.) *(KDD '03)*. Association for Computing Machinery, New York, NY, USA, 137–146. https://doi.org/10.1145/956750.956769

[9] Judit Lecina-Diaz, Albert Alvarez, and Javier Retana. 2014. Extreme fire severity patterns in topographic, convective and wind-driven historical wildfires of Mediterranean pine forests. *PloS one* 9, 1 (2014), e85127.

[10] Sang-Woo Lee, Myung-Bo Lee, Young-Geun Lee, Myoung-Soo Won, Jong-Jin Kim, and Sung-kwon Hong. 2009. Relationship between landscape structure and burn severity at the landscape and class levels in Samchuck, South Korea. *Forest Ecology and Management* 258, 7 (2009), 1594–1604.

[11] Garrett W Meigs, Christopher J Dunn, Sean A Parks, and Meg A Krawchuk. 2020. Influence of topography and fuels on fire refugia probability under varying fire weather conditions in forests of the Pacific Northwest, USA. *Canadian Journal of Forest Research* 999 (2020), 1–12.

[12] Ralph M Nelson Jr. 2002. An effective wind speed for models of fire spread. *International Journal of Wildland Fire* 11, 2 (2002), 153–161.

[13] José Olabarria, Marc Palahí, Antoni Trasobares, and Timo Pukkala. 2007. A fire probability model for forest stands in Catalonia (north-east Spain). *Annals of Forest Science* 64 (02 2007), 169. https://doi.org/10.1051/forest:2007047

[14] Cristobal Pais, Jaime Carrasco, David L Martell, Andres Weintraub, and David L Woodruff. 2019. Cell2fire: A cell based forest fire growth model. *arXiv preprint arXiv:1905.09317* (2019).

[15] N. Pathak, A. Banerjee, and J. Srivastava. 2010. A Generalized Linear Threshold Model for Multiple Cascades. In *2010 IEEE International Conference on Data Mining*. 965–970. https://doi.org/10.1109/ICDM.2010.153

[16] Marie-Pierre Rogeau and Glen W Armstrong. 2017. Quantifying the effect of elevation and aspect on fire return intervals in the Canadian Rocky Mountains. *Forest Ecology and Management* 384 (2017), 248–261.

[17] R. C. Rothermel. 2017. A Mathematical Model for Predicting Fire Spread in Wildland Fuels.

[18] The USDA Forest Service. 2020. *FSTopo Map Product*. https://data.fs.usda.gov/geodata/rastergateway/states-regions/states.php

[19] Paul A Werth, Brian E Potter, Martin E Alexander, Craig B Clements, Miguel G Cruz, Mark A Finney, Jason M Forthofer, Scott L Goodrick, Chad Hoffman, W Matt Jolly, et al. 2016. Synthesis of knowledge of extreme fire behavior: volume 2 for fire behavior specialists, researchers, and meteorologists. *Gen. Tech. Rep. PNW-GTR-891. Portland, OR: US Department of Agriculture, Forest Service, Pacific Northwest Research Station. 258 p.* 891 (2016).

[20] S. Yassemi, S. Dragićević, and M. Schmidt. 2008. Design and implementation of an integrated GIS-based cellular automata model to characterize forest fire behaviour. *Ecological Modelling* 210, 1 (2008), 71 – 84. https://doi.org/10.1016/j.ecolmodel.2007.07.020