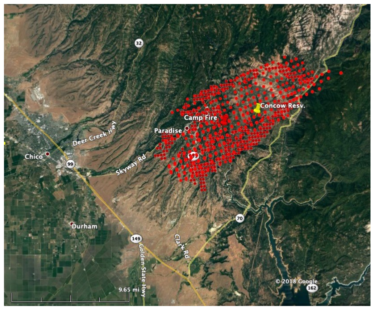
# Cellular Automata Based Mathematical Model for the Spread of Forest Fires

samyak.surti@gmail.com

April 2019

## 1 Abstract

In the United States, every year since the year 2000, there have been, on average, 73,200 forest fires that have burned a staggering 6.9 million acres of land. Within these acres of land, numerous public areas and homes burned down, displacing the lives of thousands of people, who then found it extremely difficult to assimilate back into society. Perhaps the most elusive aspect of fires is that they can be started by the most minute of factors and go unnoticed until they become too large to contain. In a situation where the fire can't be contained in an adequate amount of time, the most important consideration for firefighters is to figure out which direction would be most effective to contain the fire from. As a solution to this issue, The proposed solution is a Cellular Automata based mathematical model that will consider, holistically, factors such as fuel amount, wind patterns and topographical features to simulate them so as to accurately predict the spread of a fire. Similar models have been developed by other researchers, but they have not taken into account all of the aforementioned factors in the context of Cellular Automata based model. Traditionally Partial Differential Equation (PDE) based mathematical models are used to simulate gas movement, fire spread and other similar dynamical systems in nature. These approaches are computationally very intensive and often times very expensive to solve to fit the field data. Cellular Automata model is capable of modeling very complex dynamical systems with simpler mathematical treatise and therefore computationally friendly. Physics and field observation driven mathematical model is proposed to derive rules for the Cellular Automata. Python based computer simulation is performed to showcase several scenarios. The ultimate goal with this model is to empower firefighters to choose where it would be most beneficial to focus their resources and contain the fire as quickly as possible. Additionally, by successfully predicting the spatial and temporal spread of forest fire, evacuation can be managed more efficiently.




(a) Satellite image on Nov 8, 2018


(b) Satellite image on Nov 11, 2018

Figure 1: Camp Fire Spread in 2018

## 2 Why a Cellular Automata?

There are three mathematical models for modeling Forest Fire Spread. Some of these options include using some kind of Deep Learning approach, such as Generative Adversarial Networks (GANs) or pursuing a spread model that is based solely on real-world data and observations to formulate an equation for the rule.

**Computational Fluid Dynamics and similar finite element methods** Similar approaches have been proposed since 1972 and modified based on experimental and real-world data that is available. Based on this, underlying physical formulation is modified. In general, such approaches are computationally very challenging with large data-sets and require a much longer time to simulate.

**GANs (Generative Adversarial Networks** Recently GANs have been very popular for generating a new frame based on training performed on previous frames of a video. Therefore, GANs have the ability to extrapolate a generalized fire spread model if it is given a collection of videos of fire spreading under various conditions. By learning what to look for in these videos, the GAN can try to formulate its own version of the spread based on the characteristics of the videos it used for training. However, a key feature that GANs would lack is being able to differentiate which real-world factors are affecting and driving the spread, not to mention the amount of data required for an accurate representation to be created. Controlling a fire would require an understanding of the physical factors that are not revealed in the GAN approach.

**Cellular Automata-Based Approach** Using a Cellular Automata (CA) is advantageous for this kind of issue as it can completely be based on the real-world factors that directly contribute the fire's spread, such as fuel load, wind patterns and speed, and topographic features. A CA has the ability to integrate

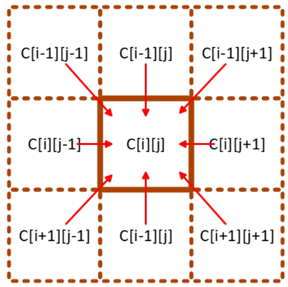



Figure 2: Moore Neighborhood

all this information and create a relatively simple rule for the spread compared to GANs or a real-world model to create a complex and dynamic mathematical model that provides and accurate representation for how fires spread.
In this work, Cellular Automata model is formulated as described in the subsequent sections.

# 3 Mathematical Background

## 3.1 Cellular Automata

A Cellular Automata(CA) is a discrete dynamical system, consisting of a homogeneous array of cells (automaton), that evolve based on a set of simple rules applied to each cell based on the states of its arbitrarily-chosen neighbors. All cells change their state simultaneously in single discrete step. CA can be 1-dimensional, 2-dimensional or multi-dimensional.

**Formal Definition of Cellular Automata(CA)**
Let, $£$ = regular lattice, (elements of $£$ are called cells) $S$ = finite set of states that a cell can be in at any given timestep $N$ = a finite set (of size n = $|N|$) of neighborhood indices such that $\forall c \in N$, $\forall c \in £$, $r + c \in £$ $f : S^n \rightarrow S$ a function, defines the transition rule. The new state of a cell is based on old states of the neighbors in N. Hence, Cellular Automata is defined as a tuple $(£, S, N, f)$.
Configuration (configuration of cells based on states), $C_t : £ \rightarrow S$ associates each cell of lattice with state for time-step t. Here Job of the function f (transition rule) is to change the configuration from $C_t$ to $C_{t+1}$. This is denoted as a transition $C_{t+1}(r) = f(\{C_t(i)|i \in N(r)\})$. Where, neighborhood in general with some distance away from r is $N(r) = \{i \in £|(r - i) \in N\}$. Specifically we use the Moore neighborhood (used in this work) is defined as $N_{i,j} = \{(k, l) \in £||k - i| \leq 1 \text{ and } |l - j| \leq 1\}$. Moore neighborhood is pictorially shown.

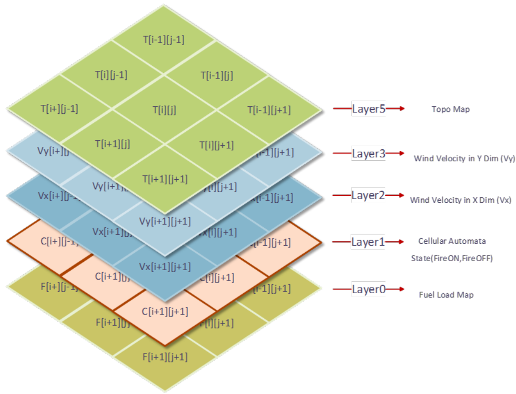



Figure 3: Layers in the proposed Cellular Automata

**Cellular Automata Layers**

**Layer 0:** Normalized Fuel Load Map, Fuel value per cell = (Total Flammable Area including factors like density, type of fuel, moisture content)/(Total Area of the Cell). This Layer is updated based if a cell is on fire.
**Layer 1:** State Map representing the actual Cellular Automata and updated every time-step. 0 indicates no fire and 1 indicates fire.
**Layer 2:** (not shown in the figure) This layer is used for temporary fuel influx computation for each cell based on $r = 1$ Moore Neighborhood, described later.
**Layer 3:** Wind Velocity in X direction in m/s. This layer can be updated at certain time-steps based on change in the wind map.
**Layer 4:** Wind velocity in Y direction in m/s. This layer can be updated at certain time-steps based on change in the wind map.
**Layer 5:** Topography of the terrain based on the average elevation of the region represented by the cell.
Following figure represents the layers in the proposed cellular Automata:

**Mathematical model for defining transition rules:**

- Basic Transition Rule is defined just based on Fuel Load Map. It is based on a gradient field from the neighbors trapped in the center cell. In the continuous space and time in 2D space, it is a line integral of gradient * normal. Here gradient is derived based on the State Map and Fuel Load Map as

  $\forall r_{i,j} \in \pounds \sum_{i=0}^{2} \sum_{j=0}^{2} ((F_{i,j} \times S_{i,j}) - (F_{1,1} \times S_{1,1})) \times A_{factor}$

  Where, $F_{i,j}$ = fuel load map value of the cell (i,j) and $S_{i,j}$ = the state map value for the cell (i,j). Here $A_{factor}$ is area factor for the fire spread. As shown in the figure, for corner neighborhood cells area contribution of the fire spread is $(\pi/4)\times$(the area of the cell). Therefore $A_{factor} = \pi/4$

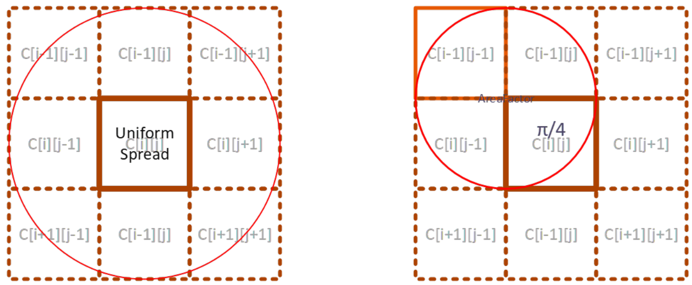



Figure 4: Area Factor

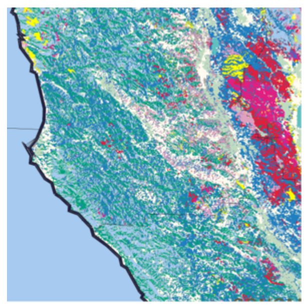


Figure 5: Real World Fuel Map for a portion of Northern California

for corner cells and 1 for adjacent cells. Following figure represents the area factor:

Following the example of the real world fuel map:

- Transition rule with Wind factor modeling is derived from vector algebra. Wind velocity is represented in 2D space over the grid as a vector $(V_x, V_y)$. $V_x$ is the wind velocity component in the X direction and $V_y$ is the wind velocity component in the Y direction. These values are stored in the Layer3 and Layer4. While evaluating the influx of fire to the center cell from the 8 neighboring cells, wind factor is derived from mathematical model of integral of all velocities' dot product with normals towards the center of the cell. In vector calculus, this in-flowing flux is computed as,

  $W = \int_c (\vec{v} * \vec{n}) ds$

  where, $W$ is a total flux flowing in to a closed curve $c$ $\vec{v}$ is the velocity vector and $\vec{n}$ is a normal at the segment $ds$ on the closed curve $c$.

  With eight neighbors there are eight normals (unit vectors) considered in

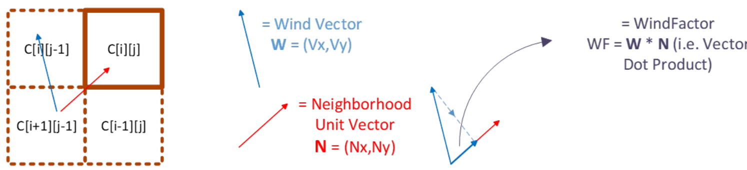



Figure 6: Wind Factor Computation

this order from upper left clock wise:
$(1/\sqrt{2}, -1/\sqrt{2})$, $(0, -1)$, $(-1/\sqrt{2}, -1/\sqrt{2})$, $(-1, 0)$, $(-1/\sqrt{2}, 1/\sqrt{2})$, $(0, 1)$, $(1/\sqrt{2}, 1/\sqrt{2})$ and $(1, 0)$.

Let's denote this normal from neighbor (i,j) as $n_{i,j}$. Similarly for cell (i,j) wind velocity vector is denoted as $v_{i,j}$. Therefore the wind factor is computed as a sum

$wind\_factor = \sum_{i=0}^{2} \sum_{j=0}^{2} (v_{i,j} * n_{i,j})$

Where * represents the dot product of two vectors. Since this is the wind-factor which gets multiplied and added to the basic gradient computed in the first step, complete fire influx is computed with wind factor is as following:

$\forall r_{i,j} \in \pounds \sum_{i=0}^{2} \sum_{j=0}^{2} ((F_{i,j} \times S_{i,j}) - (F_{1,1} \times S_{1,1})) \times A_{factor} \times (1 + wind\_factor)$

Following figure depicts the mathematical concept for the wind factor:

- Considering topographic factors, a general physical principle of how a fire propagates was considered. According to this principle, fires tends to spread up a slope faster due to shorter distance between the flame and the slope due to the fact that heat rises, causing fuel above the fire to heat up, dry out, and catch on fire first. Experimental and real-world models indicate that rate of the fire spread uphill and the incline is related in a quadratic manner. However, this principle has been recently challenged with the Carr Fire in mid-2018, when firefighters observed that fires seemed to be spreading more rapidly downhill. These fires have spread rapidly down the slope even in the absence of wind factor, as opposed to moving faster up the slope. The reason, firefighters suspect is that after winter and the massive drought in California, there is a concentration of very dry, even dead, fuel packed together, resulting in fires spreading faster downhill. Novel approach here presents this new learning in defining the transition rule with topographic factor. For each cell in the neighborhood a slope is computed as shown in the figure and slope of the sign is preserved.

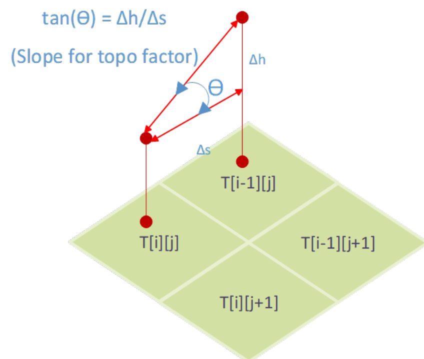



Figure 7: Slope computation for Topographic Factor

Hence, $\forall (i,j) \in N, slope_{i,j} = (topo\_map_{i,j} - topo\_map_{1,1})/distance$, where,

$$distance = \begin{cases} 1 & \text{if (i,j) is adjacent cell} \\ \sqrt{2} & \text{if (i,j) is a diagonal cell} \end{cases}$$

Following figure depicts the slope computation for topographic factor:

Topographical factor is therefore proposed based on the Fuel Load value of the current cell with a high threshold value. Let's call it $FL\_threshold$. $\forall (i,j) \in N$

$$topo\_factor(i,j) = \begin{cases} slope^2 & \text{if slope} \leq 0 \text{ and } FL\_map(1,1) > FL\_threshold \\ slope^2 & \text{if slope} > 0 \text{ and } FL\_map(1,1) \leq FL\_threshold \\ slope & \text{if slope} > 0 \text{ and } FL\_map(1,1) > FL\_threshold \\ slope & \text{if slope} \leq 0 \text{ and } FL\_map(1,1) \leq FL\_threshold \end{cases}$$

Based on all of the above three mathematical models and similar to Rothermel's model, superposition of these factors, the complete influx of fire is defined as follows:
$influx = \forall r_{i,j} \in \pounds \sum_{i=0}^{2} \sum_{j=0}^{2} ((F_{i,j} \times S_{i,j}) - (F_{1,1} \times S_{1,1})) \times A_{factor} \times (1 + wind\_factor + topo\_factor(i,j))$

Finally the state transition rule is described based on a threshold of influx to determine in every time-step, if the fire spreads to the center cell. This operations are performed for all the cells in the cellular automata simultaneously.

$\forall (i,j) \in \pounds$,

$$state(i,j) = \begin{cases} 1 & \text{if } influx >= threshold \\ 0 & \text{otherwise} \end{cases}$$

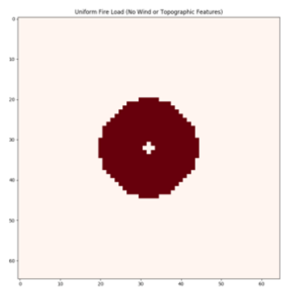



Figure 8: Uniform Fuel Load without any wind or the slope

Further, for every time-step, fuel map for each cell on fire depletes by the common $burning\_rate$.
When the cell is completely burned, fuel load value for that cell becomes zero eventually and therefore, that cell cannot be on fire.

$$FL\_map(i,j) = \begin{cases} max(FL\_map(i,j) - burn\_rate, 0) & \text{if } state(i,j) == 1 \\ FL\_map(i,j) & \text{otherwise} \end{cases}$$

## 4 Results

Python based simulator is implemented with 50X50 grid with 6 layers of the proposed cellular automata. One of the most important litmus test for fire model is to check for standard patterns with uniform fuel map. Following are the cases described to include each factor individually:

- First uniform fuel map the fire starting from the center of the 50X50 grid is implemented and simulated for 125 time-steps. The evolution of fire spread after 125 time-steps looks as shown below:
- Wind factor is added with constant wind for all time-steps with $(vx, vy) = (1, 0)$. After 125 time-steps, fire spread is evolved as shown below:
- If only slope factor is considered, the model has to perform well. Here the fuel map is uniform but it is below the FL_threshold. Slope is increasing in the south-east direction in the constant manner. After 125 time-steps the fire spread evolved to the following:
- Finally all the factors are added and the evolution of the fire is simulated over the 125 time-steps:

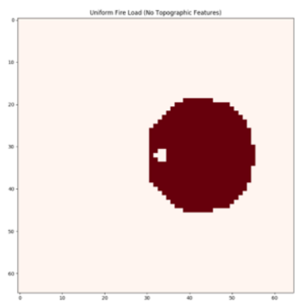



Figure 9: Uniform Fuel Load with constant wind and no slope

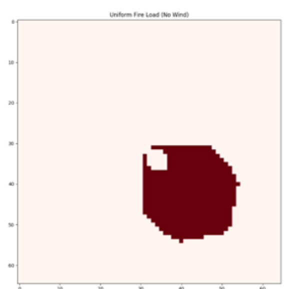



Figure 10: Uniform Fuel Load without any wind and with constant slope

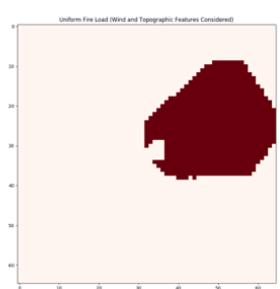



Figure 11: Uniform Fuel Load with wind and slope

## 5 Future Directions

- Along with a general rule for spreading the fire, a Deep Learning method can be employed to find out the best way in which a fire should be extinguished. A separate mathematical, Cellular Automata-based model can also be used for that same reason in conjunction with the Cellular Automata that is propagating the fire.

- Gather data from various fires that have occurred, in regards to the fuel load that was present before the fire, weather and wind patterns, topography of the region, and any other factors that may contribute to the spread of the fire, apply these same characteristics to the layers of the Cellular Automata and fine tune the parameters so that the simulated flame propagates in fashion similar to the fire in the real-world.

- In the current state of the model, the spread rate of the fire can at least be described by a first-order derivative. However, to get an even more accurate representation of how a fire spreads, developing a mathematical model that has characteristics describing the spread of the fire as second or even third-order derivatives would be ideal. Currently, since we're only considering a Moore neighborhood of radius 1 for each timestep, there is a notion of a linear spread rate, but a quadratic spread rate is not achieved in the model's current form, but is represented mathematically. Such results can be achieved through a proposed Hierarchical Cellular Automata, in which each cell of the Cellular Automata contains a CA within it.

- To address the run time of the CA, given that it is currently running exclusively on a CPU, and given that the architecture of a CA is such that it is theoretically GPU friendly, if its computations could be parallelized, it would allow for the cellular automata to run faster for a bigger lattice with a greater number of states, as well as support a Hierarchical CA, proposed above.

- After testing the accuracy of the model with multiple fires that have occurred, try talking to local firefighters to pilot the system, and get input from them in regards to what other factors may be considered, such as the fuel load of residential areas, as opposed to factors simply found in forests.

## 6 Conclusion

For a physical, dynamic system as complex as a forest fire, that is heavily dependent on a variety of different factors, a Cellular Automata did a surprisingly good job simulating the behavior of this system with relatively simple rules to define the propagation of the fire. The litmus test for my model was to see if it would spread in a perfect circle given that the fuel load levels were uniform across all cells, there was no wind, and the topography of the region was completely flat. Past this vanilla approach, adding factors such as a generalized wind pattern, fuel loads that were normally distributed, and a notion of changing topography, yielded a fairly accurate propagation of a fire. For the future, I would like to improve the accuracy of my model by considering more physical factors along with making the spread rate of my model defined as a polynomial rather than a linear function. Eventually, I would like to be able to pilot this model with firefighters and get their input on other factors I may have missed, as well as be able to propose the most efficient way to contain the fire and keep it from spreading to residential areas, saving the lives and livelihoods of many.

# References


[1] *Mapping and Estimating Forest Fuel with Radar Remote Sensing.* Fire Science Brief, 2009.

[2] A. Ganteaume. *A Review of the Main Driving Factors of Forest Fire Ignition Over Europe.* archives-ouvertes, 2013.

[3] Tiziano Ghisu. *An Improved Cellular Automata for Wildfire Spread.* Elsevier B.V., 2015.

[4] William Gilpin. *Cellular automata as convolutional neural networks.* arXiv, 2018.

[5] Wei Huang. *General Purpose Cellular Automata Programming.* Iowa State University Digital Repository, 2002.

[6] Ali Karouni. *A Simplified Mathematical Model for Fire Spread Predictions in Wildland Fires Combining between the Models of Anderson and Rothermel.* 2014.

[7] VALERIY PERMINOV. *Mathematical modeling of forest fire spread.* 2009.

[8] Richard C. Rothermel. *A MATHEMATICAL MODEL FOR PREDICTING FIRE SPREAD IN WILDLAND FUELS.* U.S. Department of Agriculture, 1972.

[9] Richard C. Rothermel. *How to Predict the Spread and Intensity of Forest and Range Fires.* U.S. Department of Agriculture, 1983.

[10] J.L. Schiff. *Introduction to Cellular Automata.*

[11] Jörg R. Weimar. *Simulation with Cellular Automata.* Logos-Verl, 1997.

[12] Stephen Wolfram. *UNIVERSALITY AND COMPLEXITY IN CELLULAR AUTOMATA.* Elsevier B.V., 1984.

[13] Mi Yu. *The Game of Life and Other Cellular Automata.* 2015.