PRINCIPLES OF COMPLEX SYSTEMS

FOREST FIRES ON DISCRETE TOPOGRAPHIES

Max Green, David Landay
University of Vermont
Graduate Students, Comp. Systems and Data Sci.

0.1 Research Question

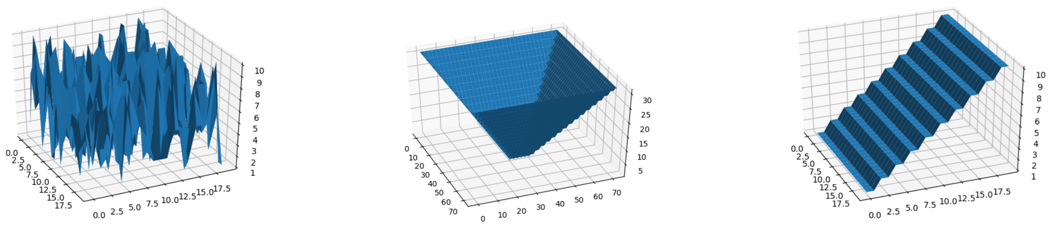
For this project, we chose to explore the forest fire model introduced in class, which examines the spread of fire over a landscape with a dual-ecology. In the original model, sites on a 2D lattice describe a multi-state substrate whereby a site can be one of four states at any given time; it can represent a tree or grass, and can either be on fire or smoldering ash. States can transition between the others under a governing set of rules, which describe how fires percolate, spread, across the landscape. This type of modelling is called cellular automata because the mechanics of state transitions over time mimic the movement of cells in a substrate.

Instead of modeling how fire spreads between two species of plants, we were curious to examine how landscapes with varying terrain effect the dynamics of the spread of the fire. Thus, we designed a 3-state cellular automata model where sites have a specified elevation and can either be trees, set on fire, or burnt ash. We then came up with a function to determine how a particular site catches fire, or transitions from tree to fire.

0.2 Modeling with Cellular Automata:

There are numerous factors that contribute to the growth of forest fires in nature. For the purposes of this experiment, we were only concerned with studying how a change in elevation between two sites on a landscape affected how sites transitioned to other states. Namely, does fire spread more quickly uphill? Thermodynamics tell us that concentrations of high energy move to areas of low energy, so it seems likely that fire - being a substance with high energy and high temperature - should spread more quickly to neighboring sites that are at higher elevations. A naiive google search, for the most part, confirms this hypothesis, so we set out to develop a model that would demonstrate the dynamics of this mechanism.

Our first step in initializing the model was to construct different topographies on which to observe the spreading of wildfires.



(a) random topography **(b)** "bowl-like" topography **(c)** slanted topography

Figure 1: Surface projections of generated topography.

Figure 1 shows examples of various topographies that we generated to observe the spread of wild fires. Each terrain is actually a 2D array of values that represent landscape elevations. By design, these arrays are the same dimension as the landscape lattices, over which we assign the states of the landscape; i.e: tree, fire, or ash. Hence for any given point in time, every site on the lattice has an associated state with an underlying height. This is visualized better in **Figure 2**. By design, each site on a lattice of dimension L x L can have a maximum height no larger than $\frac{1}{2}$ L.

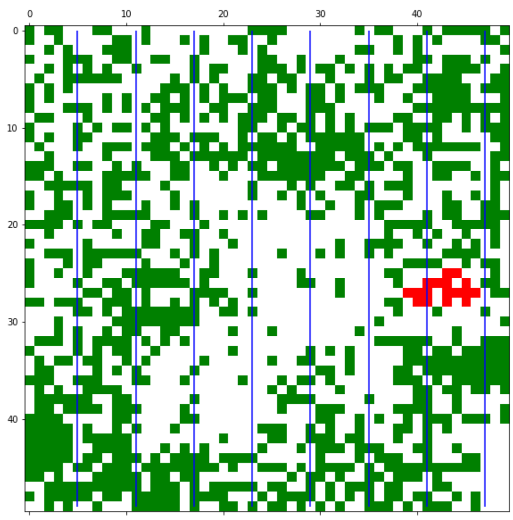


Figure 2: landscape (size 50 x 50) projected onto a "slanted" topography. Sites that are forest are represented by green pixels, sites that are burnt or not trees are white pixels, and sites that are burnt are red pixels.

Next, we had to design a rule that would govern the switching mechanics between states on the lattice. Much like the example in class, we will initially generate our landscape by placing trees at each site with some probability ρ_{tree}. We will represent sites that are trees with a "2", sites that are not trees or burnt with a "0". At time $t = 1$, we will simulate the outbreak of a fire by randomly assigning a site a state of "1". Each of the neighboring sites of the outbreak site will now have a probability p_{fire} of igniting (becoming a "1") given that they are a tree, or not burnt, associated with it, and we want to define these probabilities to favor fire that spreads up-hill. Therefore, p_{fire} at any site can now be generated by a function of the number of the site's neighbors which are on fire, and by the change in elevation between the site and its burning neighbors.

For any site that neighbors the outbreak, p_{fire} is generated by

$$p_{fire} = P(N_f, \Delta z) = \gamma + (1 - \gamma) \frac{\sum^{N_f} \Delta z \, N_f}{\mathbf{max}(z) \, N_s} \tag{1}$$

where N_f is the number of neighbors that are on fire, Δz is the difference in elevation between a site and a neighbor that is on fire, $\mathbf{max}(z)$ is the maximum height on the lattice, N_s is the total number of neighbors surrounding the outbreak (*note that this number can change depending on whether a site is at the edge or corner of a lattice), and γ represents the probability of a fire given that $\Delta z = 0$.

Over the course of time, the fire will propagate to other sites. We have modelled this both synchronously and asynchronously. We gain the most insight for statistical analysis by examining an asynchronous spread of fire, so the results section will focus primarily on the outcomes of events under an asynchronous updating scheme. The Asynchronous scheme is as follows:

randomly assign a coordinate on the lattice for fire to strike

define a set of sites that neighbor the outbreak *unvisited* $= \{(i,j)_{neighbors}\}$

While len(*unvisited*) ! = 0, **do**

  For *n* in *unvisited*

    find neighbors of n

    find neighbors on fire

    calc. sum of dz relative to n

    For $neighbors$ in n

      put only the neighbors of the outbreak in unvisited, not the outbreak itself

    calculate $p_{fire} = P(\, N_f, \Delta z\,)$

    If $randint() < p_{fire}$

      set the state of n to 1

      pop n from unvisited

      keep n in a list of fired cells

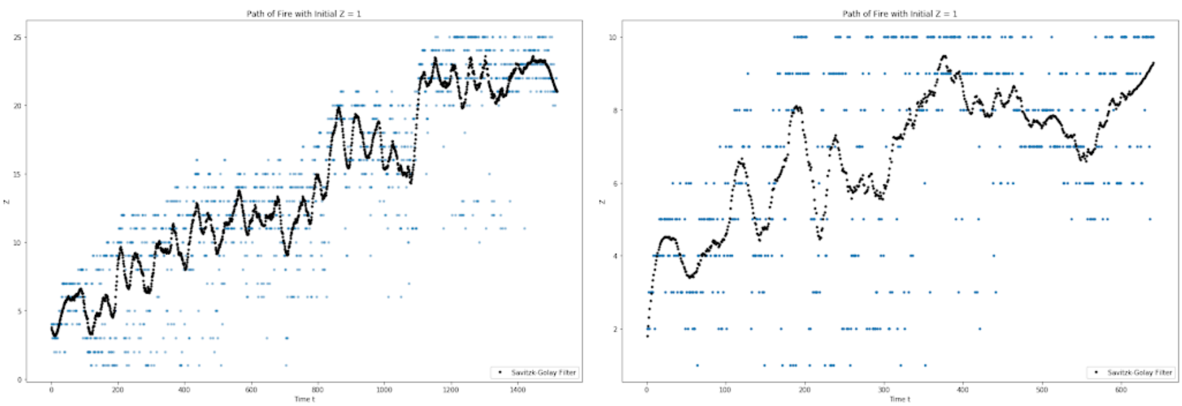
goto top

turn states of all fired cells to 0

It is worth noting that the set *unvisited* neighbors will initially pass over sites in the component that do not burn when they are not initially burned. This equates to a patch of forest, or a single site, that takes longer to ignite. We thought that updating the burning of an entire component in this manner was more true to how a fire actually spreads.

0.3 Results Discussion:

To understand how the model will respond to incentives to travel uphill, we aggregate the paths of some hundred wild fires and track the movement along the z-axis of the landscape. We would expect that on average, the fires would tend to move uphill from their origin. Fires struck in similar sized components were chosen to be aggregated. The special dynamics of the fire are largely influenced by the component structure. To relax this, we allow trees to grow in excess. This way the behavior of the fire will be left up to the probability function. In both the bowl and slanted landscapes, we see that on average, the fire moves uphill. However, the data is very noisy. A convolutional polynomial least squares smoothing function has been overlaid on top of the raw data to see the uphill trend emerging.



(a) "slanted" topography

(b) "bowl-like" topography

Figure 3: Aggregate fire movement over time

Since the entire component must burn each trial, all trees will burn, the change in elevation determines which trees get burned first.

0.4 Future Work:

There are obvious adaptations that can be made to the model. One interesting experiment we wish to conduct would be to initialize the model with real-world topography. Specifically, if we take satellite data of a real landscape where a fire outbreak has occured, we can compare the theoretical spread of a fire given known initial conditions, to an actual wild fire. In this way, we can determine how close-to-life our model is performing.

In addition to real-world tests, one clear change to make to the model is to generate continuous topographies. Currently, the discrete topographies don't allow for much variation between sites, and we get very low variance among the distribution of p_{fire} for any given landscape. We experience this less with randomly generated topographies, where the difference in heights among neighboring sites can be greater than one, but for more uniformly distributed terrain the $\mathbf{max}(dz)$ among sites and their neighbors places an upper bound on the rate at which fire can spread (i.e: when 4 neighbors are on fire). We either need more variation in the surrounding terrain at any given site, or allow continuous heights to exist. Building off of this, we also would like to alter p_{fire} to account for situations when Δz is greater than some common threshold. Obviously, in the real world, fire will spread less easier between points in space that are of vastly different heights; like when a fire is at the bottom of a cliff.

Finally, we think that it would be interesting to add wind as a tunable parameter to the model. Perhaps as a way to quell the spread of fire up hill by allowing fires to "jump" from sites of a specified distance.