CSE 6010 Assignment 1: Modeling Forest Fires

Hongyang Wang

I. Literature Survey

In applied mathematics, a forest-fire model is any of a number of dynamical systems displaying self-organized criticality.[1] The model can also be defined as a cellular automaton on a grid with L^d cells, where L is the side-length of the grid and d is its dimension, according Drossel and Schwabl.[2] So what we are expected in this simulation assignment is a grid of forest, the number of trees in which is always changing and is restricted by the growth rate and the fire rate.

Previous works have been done to address forest fire problems. Ljiljana[3] modeled forest fires using cellular automata approach. The difference with my work is that he used Moore's neighborhood instead of Van Neumann's neighborhood metric.

RM Almeida and EEN Macau[4] used stochastic cellular automata modeling for wild land fire spread dynamics. In their work there are three possible states characterizing each cell: vegetation cell, burning cell and burnt cell, which means the spreading of the fire takes some time rather than instantly. In this assignment however, there are only two states for each cell: occupied by a tree or burnt, and is assumed that the spread of the fire is instantaneous.

Instead of treating each cell as a square, it can also be other shapes. L. Hernandez[5] modeled forest fires spread using hexagonal cellular automata. This model is more realistic and the results are in agreement with the fire spreading in real forests. For this assignment, we simplified the problem by only considering square cells.

There are also some works addressing the forest fire problem by theoretical analysis. Sang Il Pak and Tomohisa Hayakawa[6] used percolation theory and mean-field approximation to derive a critical probability, below which forest fires do not expand to infinitely large area. In this assignment, only a Monte Carlo simulation will be implemented.

II. Implementation Details

1. Main function

The main function contains 3 parts:

(1) Read and parse command line arguments, including height & weight of the grid, growth rate (g), fire rate (f), number of iterations per simulation and

whether print debugging information or not. I used the command line argument so that each parameter will be determined at runtime;

(2) Run the simulation, i.e. the growth phase and fire phase, for a number of iterations (time steps), and after each iteration record the total number of trees (P) in the current forest;

(3) Calculate the average P and print the result. Here, I calculated average P per area, that is for a 100x100 grid, I got P / 10000 as my final result. In this way, P will between 0 and 1, and this result can be generalized when the grid size differs.

2. Growth phase

The simulation for growth phase is straightforward. I used a 2-D Boolean array to represent the forest. Then I iterated the array and if there is an empty spot (with false value), I generated a random number between 0 and 1 and occupied the spot (set the value to true) if the number is smaller than g. The time complexity is O(N), where N = height * weight.

3. Fire phase

I used a similar way as the growth phase to find the spot to be ignited. For the spread of the fire, I used the depth-first search algorithm and recursively burnt the surrounding tree until no connected trees to burn. The time complexity is also O(N).

III. Proof of correctness

To show the correctness of my code, I debugged on a smaller grid (10x10) with a small number of iterations. Just set the is_debug command-line argument to 1 and the debugging information will be printed. The following is some pieces of the info:

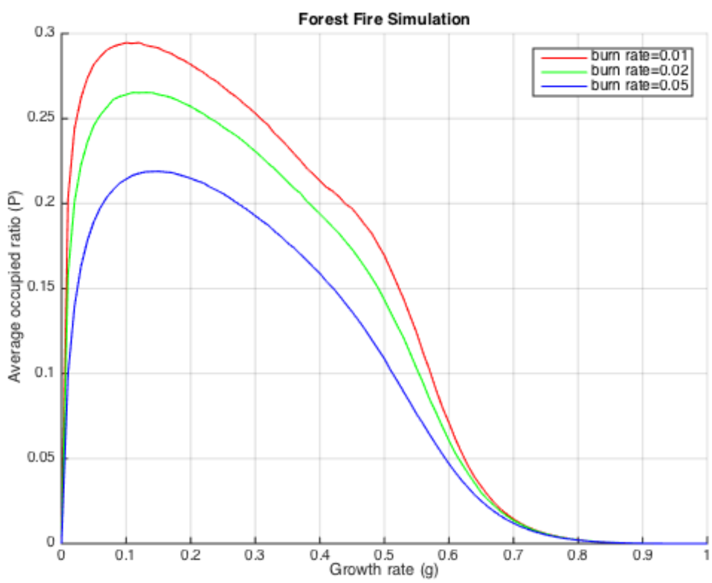
```
-----------------------------------------------------------------------------
t = 6, after growth phase:
 .  .  Y  Y  Y  Y  Y  Y  Y  Y
 Y  Y  .  .  .  Y  Y  .  Y  Y
 Y  Y  .  .  .  .  .  .  .  Y
 Y  Y  Y  .  Y  .  .  .  .  Y
 .  .  .  .  .  Y  Y  Y  .  Y
 .  .  Y  .  .  Y  Y  .  .  .
 Y  .  .  .  .  Y  .  Y  Y  Y
 .  Y  .  Y  Y  .  .  Y  Y  .
 .  .  Y  .  Y  Y  .  .  .  Y
 .  .  .  Y  Y  .  Y  Y  .  .
Number of trees = 47
t = 6, after fire phase:
 .  .  .  .  .  .  .  .  .  .
 Y  Y  .  .  .  .  .  .  .  .
 Y  Y  .  .  .  .  .  .  .  .
 Y  Y  Y  .  Y  .  .  .  .  .
```

```
 .   .   .   .   .   Y   Y   Y   .   .
 .   .   Y   .   .   Y   Y   .   .   .
 Y   .   .   .   .   Y   .   .   .   .
 .   Y   .   Y   Y   .   .   .   .   .
 .   .   Y   .   Y   Y   .   .   .   Y
 .   .   .   Y   Y   .   .   .   .   .
Number of trees = 25
----------------------------------------------------------------------------
```

As can be seen from the forest before and after the fire, if some tree burnt then all its neighbors get burnt, and finally all the trees in the connected region get burnt. This proves the correctness of the DFS part, which is the most error-prone part of the simulation.

IV. Experiment result

Since the main function will only run the simulation for one set of (g, f) each time, I wrote a shell script to conduct a set of experiment. For g, it ranges from 0 to 1 included with a step of 0.01 for each f. Finally I used Matlab to plot the graph:



As can be seen from the plot, as g grows, the average occupied ratio (P) first goes up and reaches the max point (which is less than 0.3) at some point around 0.1. Then the curve goes down and finally shrinks to 0. For different f's, P decreases as f grows.

This result makes good sense. When the forest is relatively sparse, with a larger growth rate more trees will grow up. However, when the growth rate is too large and the forest becomes dense, if there is a fire at some point it's highly likely a great number of trees will get burnt. For the fire rate, it's obvious that the forest will get smaller if more trees are prone to catch a fire.

V. Reference

[1] Wikipeida. https://en.wikipedia.org/wiki/Forest-fire_model
[2] Drossel, B. and Schwabl, F. (1992), "Self-organized critical forest-fire model." Phys. Lett. A 149, 207-210.
[3] Bodrozic, L., Stipanicev D. and Seric M., Forest fires spread modeling using cellular automata approach. Kosice, Slovakia: equilibra, 2006. 23-33.
[4] Almeida, R.M. and Macau, E.E.N.: Stochastic cellular automata model for wildland fire spread dynamics, J. Phys. conf. Ser., 285, 012038, doi: 10.1088/1742-6596/285/1/012038, 2011.
[5] L. Hernandez Encinas, S. Hoya White, A. Martin del Rey, G. Rodriguez Sanchez, Modeling forest fire spread using hexagonal cellular automata, Appl. Math. Model., in press. 2006.
[6] Paki, S.I., Hayakawa, T.: Forest fire modeling using cellular automata and percolation threshold analysis. In: Proceeding of the American Control Conference, art. no. 5991603, pp. 293-298. 2011.