# CSS project

nickhatzis

June 2022

## 1 SOC in the virus propagation model

### 1.1 H-I-R-E-D model

Our virus propagation model is constructed on the assumption that nodes operate on 5 states as follows:

1. state 0, where the node is Healthy
2. state 1, where the node is Exposed: has the virus and can spread it, but won't die from it
3. state 2, where the node is Infected: has the virus and can spread it, but also can die from it
4. state 3, where the node is Recovered: was exposed or infected and survived, has immunity from getting the virus again in this state
5. state 4, where the node is dead

We provide a visualization here, with the assumption that Healthy can only change state if a neighbor is Exposed or Infected:

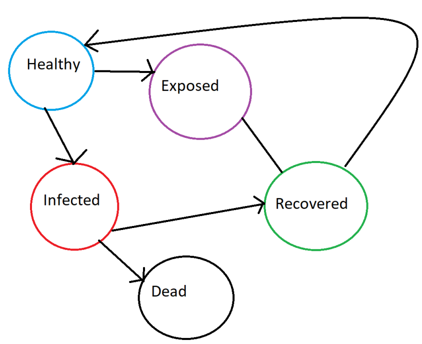



Figure 1: The initials of each state fit the name of the model

This characterization naturally fits the SOC model as follows:

1. The model is in a stable state, where all nodes are healthy, ie. state 0
2. Some disturbance is thrown on the system with the introduction of some infected nodes
3. there is an avalanche caused where nodes change between states
4. finally the model reaches some new equilibrium state where some nodes are dead and some healthy, or all nodes are dead

Repeated applications of disturbance will reduce the model to the "all dead" state with probability 1. This is easily predictable and not very interesting. We will focus instead our investigation on the following questions. However before we present our question we introduce some terminology:

1. Avalanche size = time after disturbance that the system takes to reach a steady state, this can be either "all dead" or "some alive, some recovered, some dead" state. But in either case no exposed or infected nodes. Note that this is our definition of avalanche size, someone else might define steady state, a state with only Healthy and Dead nodes (not Recovered)
2. "Energy in" is measured as per number of infected nodes introduced into the system
3. "Energy out" is measured as per number of dead at the end of the trial
4. In this model we implicitly assume that the probability of a dead node to come back to life is zero

## 1.2 How does avalanche size increase with network size?

The easy assumption here can be that the avalanche should increase as the network increases in size. The logic follows, more nodes $\rightarrow$ more time that they gonna jump around from state to state, hence avalanche size increases.
Another train of thought could be, that in a bigger network, the infection increases exponential faster (due to number of links between nodes, this depends on the network itself). In this case, we might actually see that the avalanche size slows down, maybe even halts increasing as the network grows.
For example, in a Barabasi-Albert model, we expect the central hubs to get infected and die fast$\rightarrow$ more clusters appear faster and this helps stop the spread, so avalanche might be lower be lower than other models. Speaking on said models, the random graph model is more well connected and robust to links dying. This might lead to spread taking longer time and increasing avalanche size.

### 1.3 How does output energy varies?

Contrary to avalanche size, energy output should increase with network size and also with connectivity (higher connectivity=more dead). The million dollar question is how are they related, ie. is the growth linear ? exponential ?
Also in contrast with the avalanche size, in an Barabasi-Albert model, central hubs are very likely to get infected and spread the virus → increasing casualties. In that sense, an Barabasi-Albert model might induce higher energy output than let's say a random graph.

## 2 How we will structure our experiments

We will analyze results, starting from early results to higher more complicated plots. However, we will follow the same guidelines, where if changed, we will explicitly mention:

1. We will generate a network and that will remain fixed for the experiment.
2. For any statistic we want to measure, we will do 100 tests and either provide an average or provide the results of the trials

For example we might construct a Barabasi-Albert network of (53,3), representing a network of 50 nodes where the final 3 are the disturbance introduced as infected nodes (something like throwing sand grains on a pile of sand). Then we measure output energy for example

We decide on some token probabilities for the model transition function (remain fixed for all experiments) and we present here:

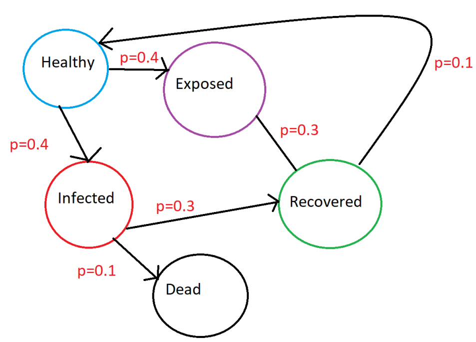
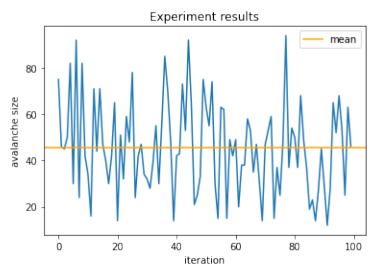


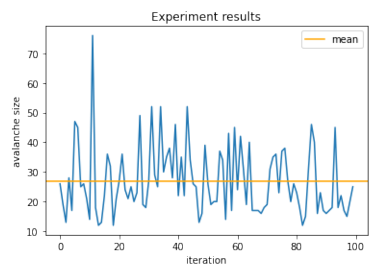

Figure 2: Our model

## 2.1 Experiment 1: Calculating the avalanche size, and energy output of a small network

In this warm up experiment we calculate the avalanche size and energy output of a (53,3) Barabasi-Albert and also the quantities for a (53,3/53) Random Network.
The choice of (53,3/53) Random Network is justified from the fact that like in Barabasi-Albert model the nodes added at the end will have degree 3
We begin with avalanche size:

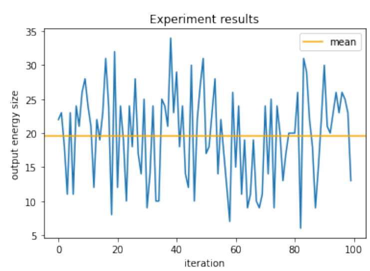



(a) Barabasi-Albert, mean= 43.81

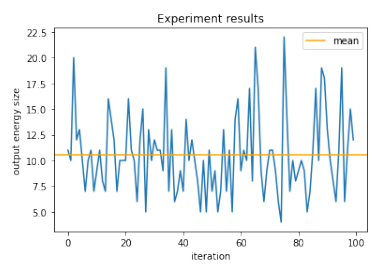



(b) Random Graph, mean= 26.93

We proceed with Energy output:




(a) Barabasi-Albert, mean= 19.61




(b) Random Graph, mean= 10.58

We note that Barabasi-Albert model has higher averages in this early network, however in the next section we will discover a more involved relation.

### 2.2 Experiment 2: grow the network

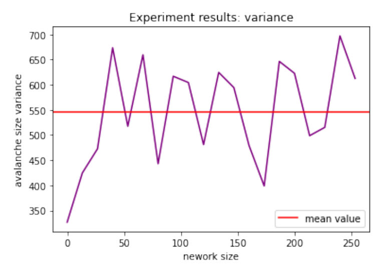
For this section we start our networks at 50 nodes (+3 as disturbance), and we increase the size of the network by 10 at each step. At each step we measure avalanche size and energy output and plot their mean values as the network size increases

To produce these subgraphs for each step, we proceed as follows. We plot a large network, and then for each n, we take the first 50+n*10+3 nodes from to measure our statistics (the +3 nodes at the end are the disturbance induced).

In particular we will use a Barabasi-Albert(253,3) network and also a Random Network(253,3/253). The orange plots on the left are a plot of the mean value of the statistic at each step (remember at each step we run the model 100 times). The purple plots on the right, track how the variance of the statistics (in these 100 trials) evolves over time. We visualize our results.

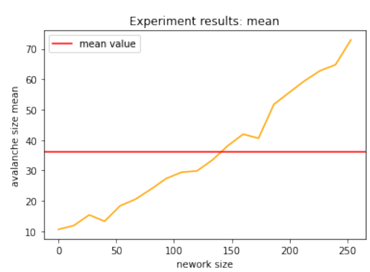
We begin with avalanche size, Barabasi-Albert:


(a) Barabasi-Albert, mean= 82.82

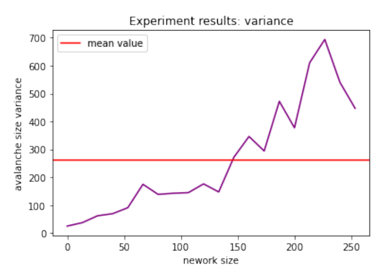



(b) Barabasi-Albert, mean= 545.53

Avalanche size, Random Graph:

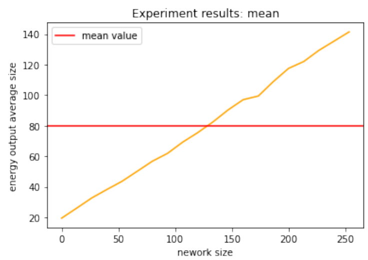



(a) Random Graph, mean= 36.10

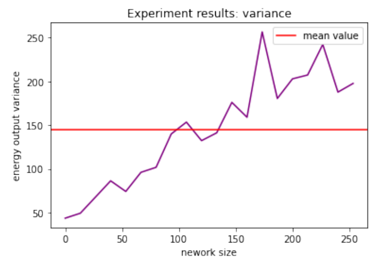



(b) Random Graph, mean= 262.94

Now with the energy output, Barabasi-Albert:

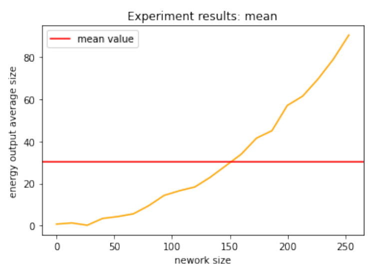



(a) Barabasi-Albert, mean= 79.85

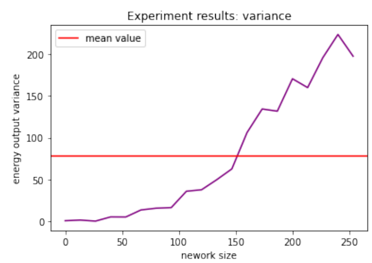



(b) Barabasi-Albert, mean= 144.82

Energy output, Random Graph:




(a) Random Graph, mean= 30.23




(b) Random Graph, mean= 78.13

## 2.3 Discussion and Logarithmic trend

The Albert-Barabasi model shows a logarithmic growth in comparison to Random Graph in both avalanche size and energy output. In particular we see that the avalanche size in Barabasi-Albert follows a logarithmic shape, while the Random network follows a linear shape. More cleanly, on output energy graph, Barabasi-Albert is linear, while Random Network is exponential.
We note that despite the above observation, the raw numbers of Random Graph are lower than Barabasi-Albert in both cases. This begs the question, what if we increase the network even more, will Random Network overtake the raw numbers ?
We extrapolate the networks (same parameters) to 1000 nodes and present the findings here:

**Extrapolate network to 1000 nodes**

| | Albert-Barabasi: | Random graph: |
|---|---|---|
| Avalanche mean: | 162.3 | 203.5 |
| Output energy mean: | 651.6 | 811.8 |

Figure 9: We see that Random Network overtakes on both values

# 3 A cellular Automata game

We introduce a cellular Automata game to study how far a virus spread. This model will very big simplification of real world processes, but it might hide insides non-the-less.
We plot an NxN 2D grid and we let the very middle square to start as infected, each cell is looking the diamond around (von Newman model) and the rules are as follows (with 5 states):

1. A healthy cell detects infected neighbors and can, stay healthy, disconnect from the network or become infected (change only happens if an Infected neighbor is detected)

2. An infected cell can stay infected, recover or die

3. A recovered or dead cell stay on that state for the rest of the game

4. A disconnected cell can remain disconnected or become healthy again

The game ends if the infection reaches the edges of the grid (lose condition), or if a specified number of turns has passed (win condition).
We decided on some token probabilities which remain fixed for the next two experiments, we visualise the model here:

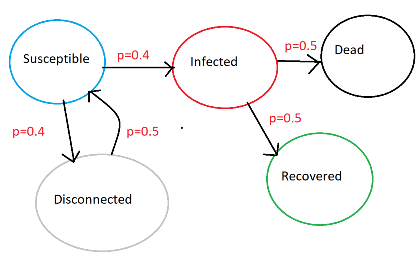
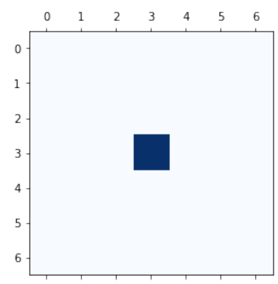


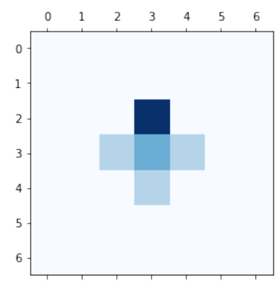

Figure 10: Our model

## 3.1 Warm Up

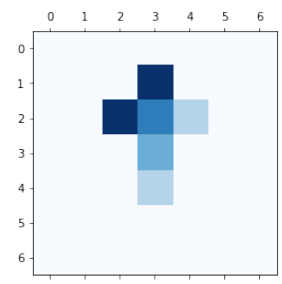
This is a simple run of the game to showcase the rules. The game runs on a 7x7 grid
The grid is color coded as follows:
white=Susceptible→Disconnected→Recovered→Dead→Infected=dark blue



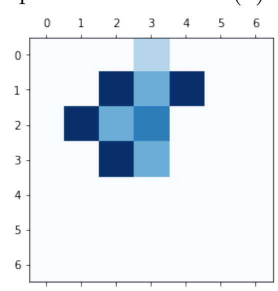
(a) step 1



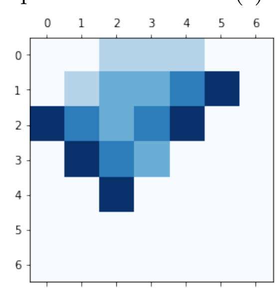
(b) step 2



(c) step 3



(d) step 4



(e) step 5, we lost

We see at picture 5 an infected (dark blue square) touched the boundary, so we lost

## 3.2 Varying grid size

In this section we start at grid of side length 21 and increase it by 2 on every iteration. At each grid size 10 experiments are run and we note the average win percentage:


Figure 12: grid size is slowly increasing

Surprisingly the increasing grid size does not seem to reduce the win percentage. However we actually want to test if win percentage indeed stays at one after grid size of 70 or it drops again like after around grid size 55. We make a new plot, studying the grid size from 51 to 251 with +10 side length increase at each step (each step is an average of 20 iterations) The probability seems

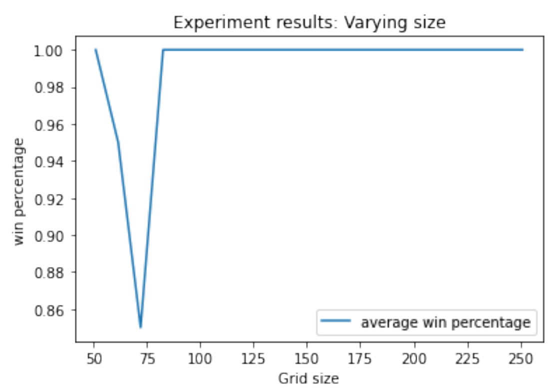



Figure 13: grid size is slowly increasing

to indeed converge to 1 after the 80 point, and surprisingly the win percentage drop from 50 to 70 is detected by both models, but we cannot attribute it to something specific.

### 3.3 Varying infection probability

In this experiment, the grid remains fixed at 51x51 and we alter the probabilities as follows:

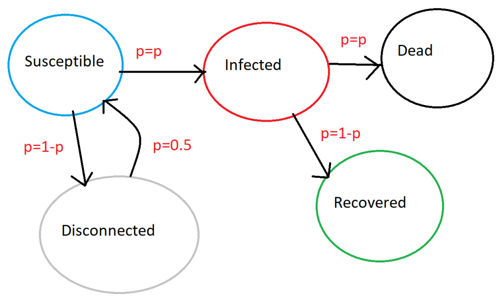



Figure 14: p is slowly increasing

The variable probability p starts at 0 and increases by 0.05 all the way until p=1. For each value of p we run the model 20 times and we note the average win probability. We get the following results:

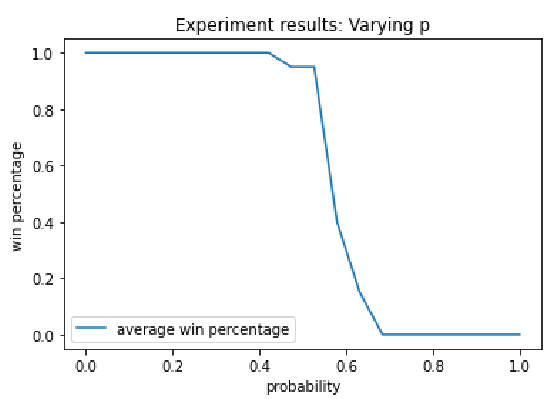



Figure 15: p is slowly increasing

This model showcases a strong hypothesis that the underlying mechanism is a phase transition around the value 0.5, that changes our win chance from 1 to 0. In particular we don't see any gradual decrease that ends with a data collapse, just a fast phase transition.

# 4 Power Law in Random and Scale-Free networks

## 4.1 Experiment 1: infected/offline nodes vs cluster size

shows a powerlaw distribution if timesteps 30

## 4.2 Experiment 2: Spyware Antivirus

Simulate 50x what happens with antivirus adoption. Research: If software isn't updated when X fraction of nodes are (unnoticed infected), then a spread will occur given these circumstances.

Results: Smaller networks adapt antivirus software later, because spyware often gets not notices in smaller networks because less nodes are infected in general.

## 4.3 Experiment 3: Purposeful targetting: Spreading probability in Barabasi-ALbert networks

Backed up by paper stating that in 95 percent of the chances, malware won't spread if its in a random, non-important computer in a barabasi-albert like network. However, if a network is purposefully target (i.e. node with greatest degree/betweenness/closeness centrality) spread is likely to occur all the time. We show this by simulating 50 times non-purposeful targetting and targetting of central nodes.
How to proof: Simulate 100x the spread in a barabasi albert network. does it spread after x timesteps?

## 4.4 When does a spread occur?

# 5