Game of Gossip

Computer Science + Computational Biology (Bonus 🙂)

Karni Abell (CB - 330292632), and Hod Bin Noon (CS - 203218037)

Definitions and assumptions

The neighbors of a cell are the eight cells that surround it.

Suggestibility/skepticism level:

- S1 - P = 1, believes everything
- S2 - P = ⅔, believes in ⅔ of the cases
- S3 - P = ⅓, believes in ⅓ of the cases
- S4 - P = 0, believes nothing

Timeline of a generation:

1. We already know who decided to gossip (the gossipers of the last generation)
2. Gossipers pass the gossip to their neighbors (candidates)
3. Candidates follow the transition rules and determine if they become gossipers

Assumptions:

The initial gossiper always gossips (regardless of their suggestibility level).

Part 1 - Game simulator

Running

The game simulator is attached to this report (source code is [on Github]()).

To run the simulator:

1. Unpack the archive
2. Run: python3 serve.py
3. Open the address that the script outputs (e.g., http://localhost:53902/)

The script is a simple HTTP server that runs a pre-built web app (in the build/ directory).

Operating

You can set P, L, and MaxGen, but you'll need to restart the game for the new parameters to take effect.

The simulator has two operation modes:

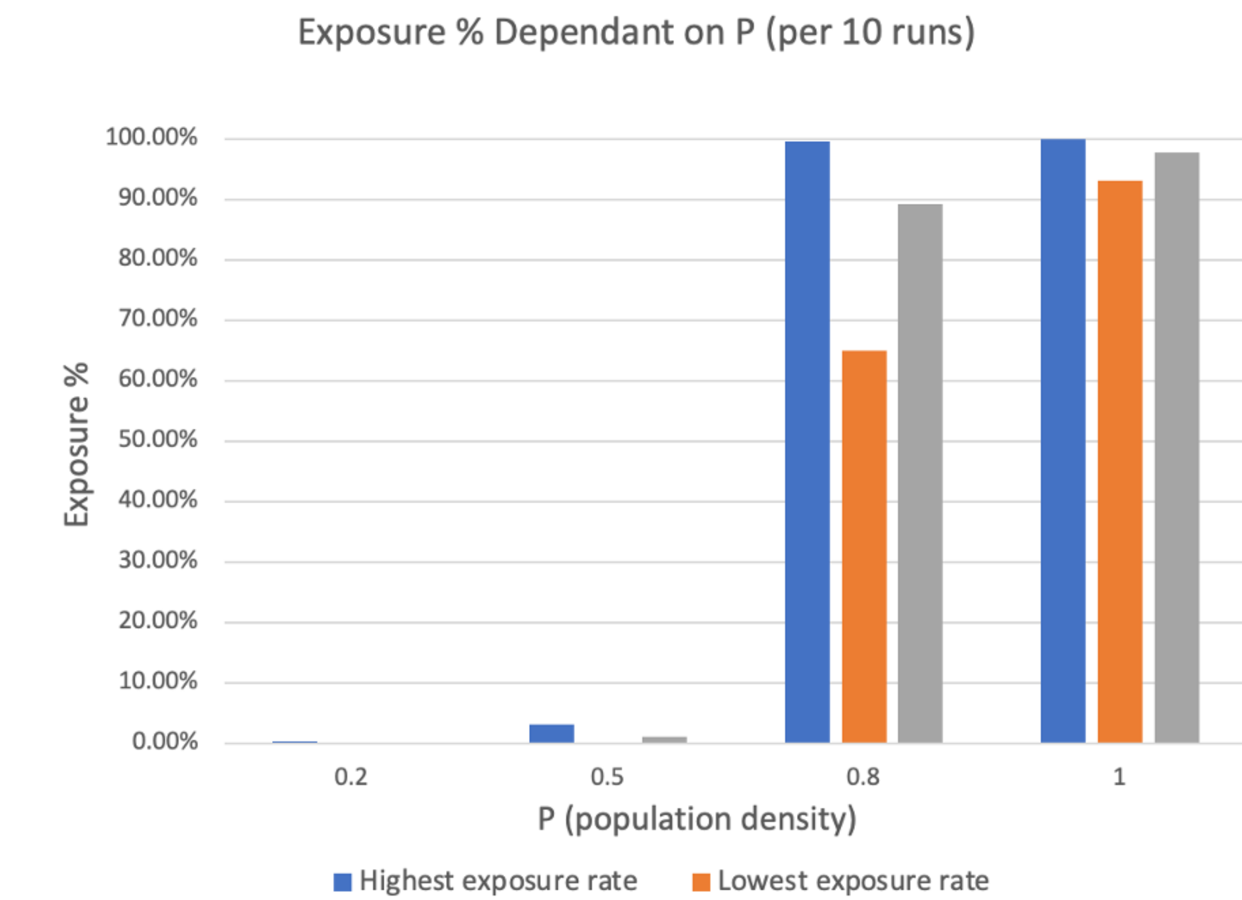
- Single step - simulates a single generation and then pauses
- Run - Run until the game ends or the simulation reaches the maximum generation. You can configure the interval (e.g., 100ms, 500ms) but be sure to re-run after you change it.

You can inspect the message log or the last game result log to see a JSON representation of the result.

Part 2 - Analysis

We analyzed how the different parameters affected the rumor spread. In figure 1 we show a chart that represents the relationship between rumor exposure percentage and the population density, P, when L is constant at 3. We ran the simulator 10 times for each P value.

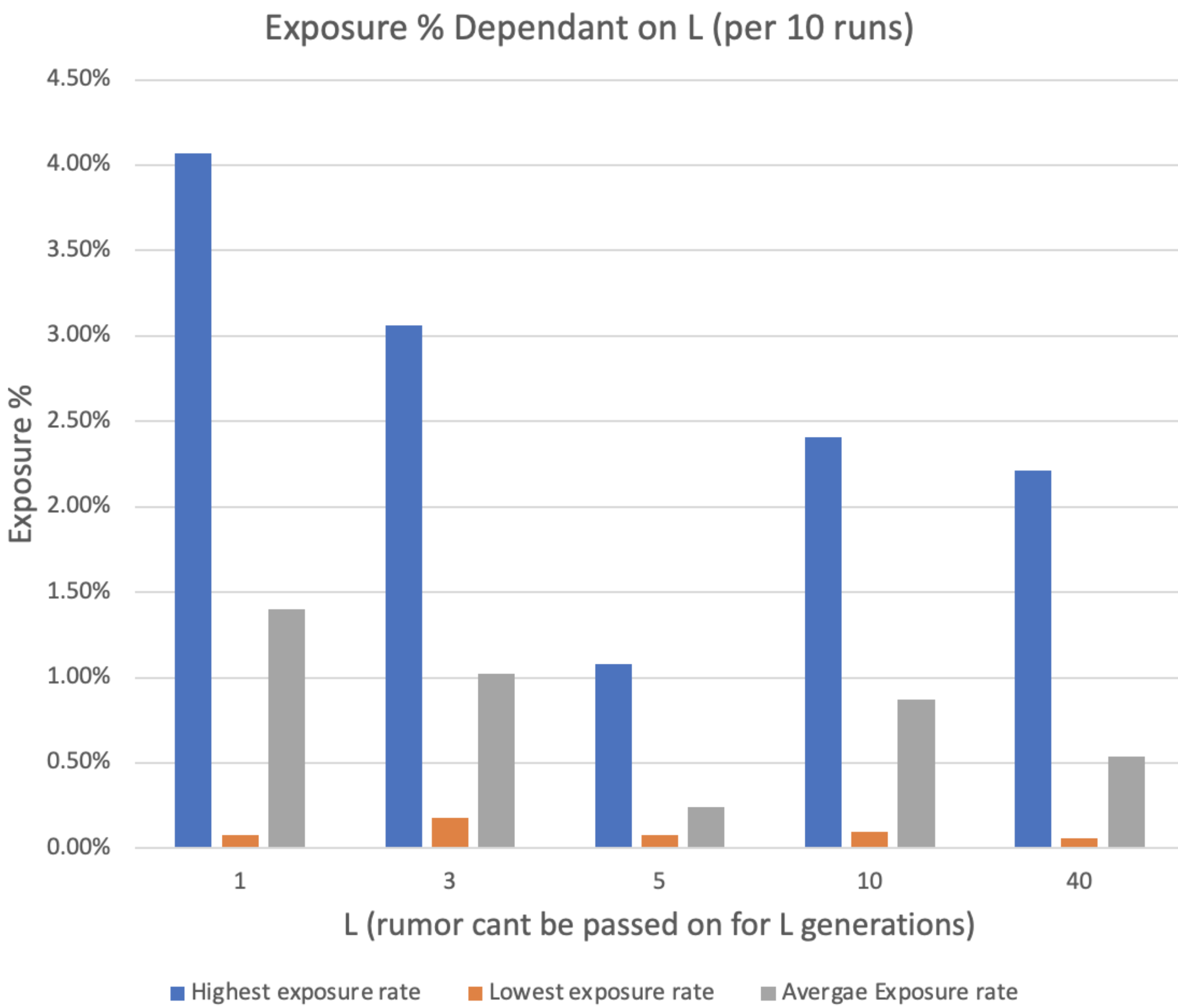
Figure 1:



As we can see, the higher the value of P, the higher the rumor exposure. This makes sense intuitively because a higher population density means that there are more people in close proximity to each other, making it easier for the rumor to spread quickly through the population. This is reflected in our results, where the exposure rate increases sharply as P increases.

Figure 2 represents the relationship between the rumor exposure percentage and the length of time (L) for which a person who has heard a rumor will not pass it on again when P is held constant at 0.5. We ran the simulator 10 times for each L value.

Figure 2:

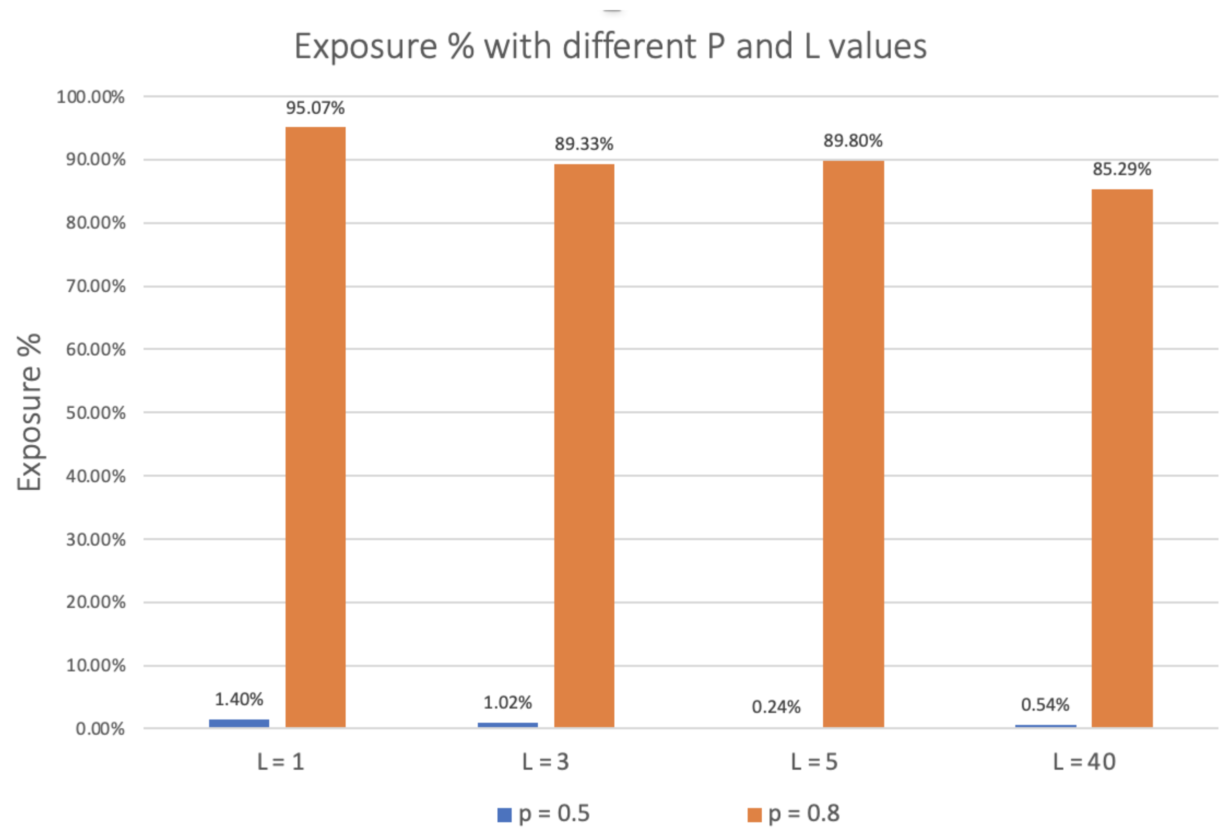


In this figure we see that the exposure rate does decrease as L grows, but not as drastically as P affects rumor exposure.

In the next chart, figure 3, we show the average exposure percentage for 10 runs dependent on P and L which can help give us a broader picture of what gives the most effect.

To get a broader picture of the effects of P and L on the average exposure percentage, we again ran the simulation 10 times for a few combinations of P and L and plotted the results in figure 3. The chart shows that while L does have some impact on the exposure rate, it is not as significant as the impact of P. As P increases, the exposure rate increases dramatically, while L has a more limited effect.

Figure 3:



We see that as L grows, the exposure percentage decreases slightly. But when P is higher we see a very drastic change, when comparing p=0.5 to p=0.8. As we see in the chart, both L and P affect rumor exposure but the effect of L is minimal compared to P.

Overall, our results suggest that population density (P) is the key factor that affects the spread of rumors. The effect of L cannot be ignored entirely, as a higher L value can slow down or even stop the spread of the rumor because too many people will not agree to spread the rumor. It is also worth noting that the behavior of the system is non-linear, as seen in the sharp increase in exposure rate with increasing P.

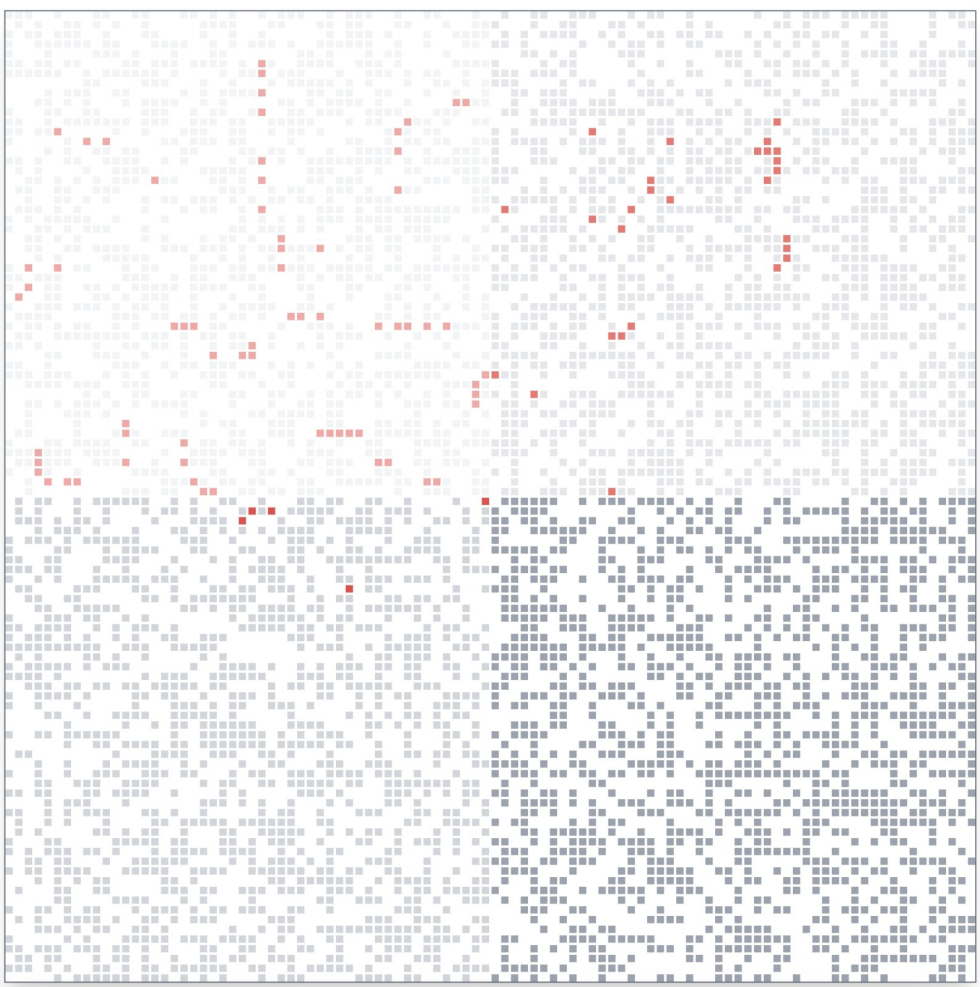
- The reason we see there is a slight dip when L=5 and then it slightly rises when L=40 is most likely because we only did the experiment on 10 runs. A bigger batch should reflect the results more accurately.

Part 3 - Suggested strategies

We suggest two strategies to lower the rumor spread rate.

The first strategy we thought of that would lower the rumor spread rate is using clustered distribution. Instead of randomly distributing the people with different levels of skepticism, we can cluster them together based on their skepticism levels. As shown in figure 4, we created four distinct regions on the grid, each representing one level of skepticism. Within each region, the population density can still be uniform. This way, people with similar levels of skepticism will be closer to each other, making it harder for the rumor to spread across regions with different skepticism levels.

Figure 4:



When we ran the simulator this way, we saw the exposure rate greatly decreased. This is because the rumor gets "stuck" and doesn't reach the S4 part of the grid.

This made us realize that if we place the S4's strategically, we can reduce the spread of the rumor even more so. As shown in figure 5, we decided to place the S4's in vertical lines. This way, the rumor spreads only by height and not by width because there are S4 "fences" blocking it from going through to the next layer. The rumor rate decreased drastically. This is because the S4 individuals have a zero probability of believing the rumor, so if the rumor cannot spread to them, it cannot continue to spread beyond their column. By placing the S4 individuals strategically in vertical lines, the rumor is essentially blocked from spreading to other columns, leading to an even more significant reduction in the overall rumor spread rate.

Figure 5: