SWEN90004 Assignment 2 Report

Ho-Yu Cheng 1063950
Ruiming Xie 1050088
Zixuan Liu 991414

Word count: 1499

1 Background of the model

1.1 Overview

We are trying to replicate the Rebellion model from the NetLogo Model Library. It is based on an agent-based computational model (Epstein [Epstein], 2002) that studies the core dynamics of decentralized upheavals against the central authority. It can be of great value for academic research in the area of Social Science and provide insight for policy makers on the stability of society.

1.2 System Parts

This model consists of a discrete Cartesian coordinate system representing the map, and an arbitrary amount of actors on it.

The model has three global values, L, representing the legitimacy of the central authority, J_{max}, meaning the maximum jail time, T, defining the threshold for agents to be active.

There are two categories of actors:

- Cops (Homogeneous)
- Agents (Heterogeneous)

Each agent has three states, $\{Q, A, J\}$, with Q representing being quiescent, A active and J in jail.

Each agent has two local values, $H \in \{0, 1\}$, representing its perceived hardship, and $R \in \{0, 1\}$, representing its risk aversion.

1.3 System Structure

Cops will enforce active agents, putting them in jail.

1.4 System Behaviour

Each actor has the same vision distance. They can only see other actors within the the radius of its vision distance. Each free agent will randomly move to an empty point within its vision if there's any, and then determine if it should make a state transition based on the value of $G - N - T$, where:

$$G = H(1 - L)$$
$$N = R * (1 - exp[-k(\frac{C}{A})]),$$

where C and A are the number of cops and active agents within its vision respectively

If $G - N - T > 0$, switch its state to A, otherwise, to Q.

Each cop will randomly pick one active agents within its vision and arrest it, generating a random discrete jail time from $U(0, J_{max})$ for this agent.

Each agent in jail will do nothing unless it has served its time and then switches to Q.

2 Our design and extension

2.1 Model Design and Implementation

1. Final UML design diagram

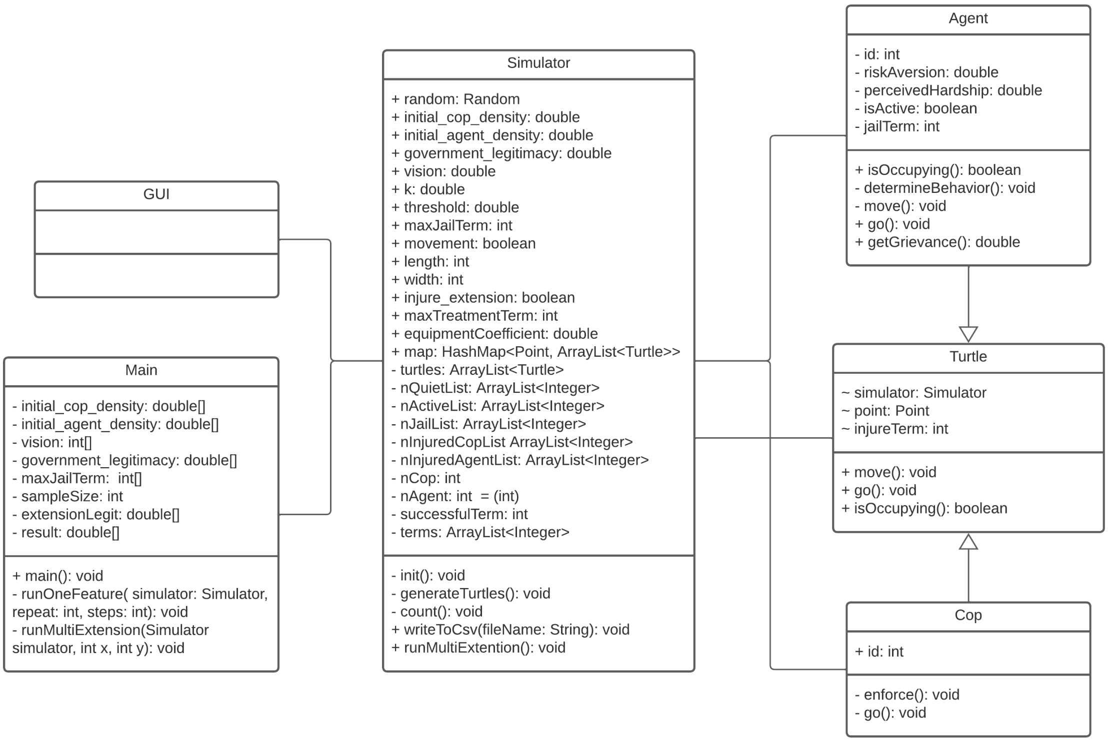

Figure 1: UML of our replication

2. Implementation process

   (a) We created a UML diagram at the proposal stage.

   (b) We used Github for version control and created a repository for the project here.

   (c) We created a template without actually implementing any method.

   (d) We implemented the model by printing out variable values for debugging at first. Then we decided to make a GUI for debugging and visualization.

   (e) We implemented methods to run multiple times with different parameters.

   (f) We extended our model for research.

   (g) We found our code is not well contained in multiple versions and rewrote Rebellion v2.0 to simplify logic and optimize algorithms for grid calculations.

2.2 Data Format

Output CSV file format of our replication:

time	quiet	jail	active
0	1120	0	0
1	1087	10	23
...	...	...	...

Output CSV file format of our extended model when using GUI to debug:

time	quiet	jail	active	injuredCop	injuredAgent	RebellionSuccessfulTurns
0	1120	0	0	0	0	0
1	1087	10	23	0	0	0
...	...	...	...	...	...	...

Output CSV file format for experiments:

legit	equipment
0.0	17.6639
...	...

2.3 Extension of our model

In the Simulator class, we added some more attributes:

- `injuryExtension`: extension mode on or off
- `maxTreatmentTerm`: the maximum number of days to stay in hospital after a fight
- `equipmentCoefficient`: with equipment (suit, guns etc.), the number of active agents that one cop can beat

In the Cop class, we adjusted the `enforce()` method so that in every arrest operation, the active agent may fight back. When a fight occurs, the weaker side goes to the hospital and receives treatment for `random.nextInt(maxTreatment+1)` days.

If the injured object is an agent, it still needs to go to jail after being cured.

3 Experiment

3.1 Research Question

If physical violence is introduced into our system and active agents are able to fight back, how well equipped do the cops have to be in order to maintain the stability of the society given different values of L?

3.2 Experiment Design

3.2.1 Model Comparison

We first designed six experiments to test if our replication has similar behaviors to the NetLogo model. For each set of parameters, we repeated 50 times. For example, we ran 50 times for experiment 1 but 250 for experiment 2, because there are 5 sets of parameters in experiment 2. We let the model run 1000 turns each time.

	Cop Density	Agent Density	Vision	Gov. Legitimacy	Max Jail Term
1	default(0.04)	default(0.7)	default(7)	default(0.82)	default(30)
2	[0.01, 0.02, 0.06, 0.08, 0.1]	default	default	default	default
3	default	[0.4, 0.5, 0.6, 0.8, 0.9]	default	default	default
4	default	default	[2, 4, 6, 8, 10]	default	default
5	default	default	default	[0.2, 0.4, 0.6, 0.8, 1.0]	default
6	default	default	default	default	[0, 10, 20, 40, 50]

The first experiment allows us to measure the dissimilarity between our replication and the original model with default parameters. The rest of them allow us to observe how our replication reacts to different parameters and compare it to the original model.

3.2.2 Research on extension

After making sure both models produces similar results, we then designed experiments to answer our research question. Before we started, we made some assumptions and definitions:

1. When being enforced, each agent decides if it wants to fight back based on its own G
2. When a fight occurs, the result is based on the number of cops times `equipmentCoefficient` and the number of the active agents within vision
3. No death from the violence.
4. If the situation where more than half of the cops are injured continues for more than 10 turns, cops failed.
5. If cops succeed 9 times out of 10, we say the cops are well-equipped enough.

We planned to run 100 sets of experiment, each with one unique value of L uniformly distributed between 0 and 1. For each set, we used an algorithm to let the model keeps running until a good approximation of the minimal `equipmentCoefficient` is found. For performance issues, we only gave 100 turns to each run. The algorithm is as below:

$$
\begin{array}{l}
total \leftarrow 10 \\
upperEquipment \leftarrow 100 \\
lowerEquipment \leftarrow 0 \\
\textbf{while } upperEquipment - lowerEquipment > 0.001 \textbf{ do} \\
\quad success \leftarrow 0 \\
\quad i \leftarrow 0 \\
\quad \textbf{while } i < 10 \textbf{ do} \\
\quad\ \ \textit{run simulation here} \\
\quad\quad \textbf{if } Rebellionsuccessful \textbf{ then} \\
\quad\quad\quad success \leftarrow success + 1 \\
\quad\quad \textbf{end if} \\
\quad\quad i \leftarrow i + 1 \\
\quad \textbf{end while} \\
\quad \textbf{if } success/total > 0.1 \textbf{ then} \\
\quad\quad lowerEquipment \leftarrow minEquipment \\
\quad\quad minEquipment \leftarrow minEquipment + (upperEquipment - minEquipment)/2 \\
\quad \textbf{else} \\
\quad\quad upperEquipment \leftarrow minEquipment \\
\quad\quad minEquipment \leftarrow minEquipment + (lowerEquipment - minEquipment)/2 \\
\quad \textbf{end if} \\
\textbf{end while}
\end{array}
$$

3.3 Experiment Result

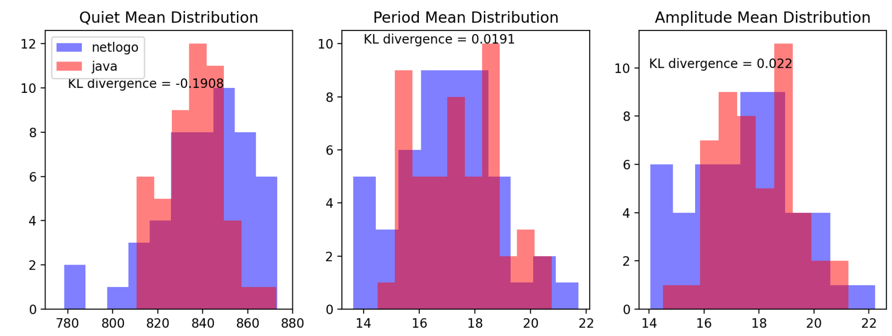

Figure 2: Comparison with default parameters

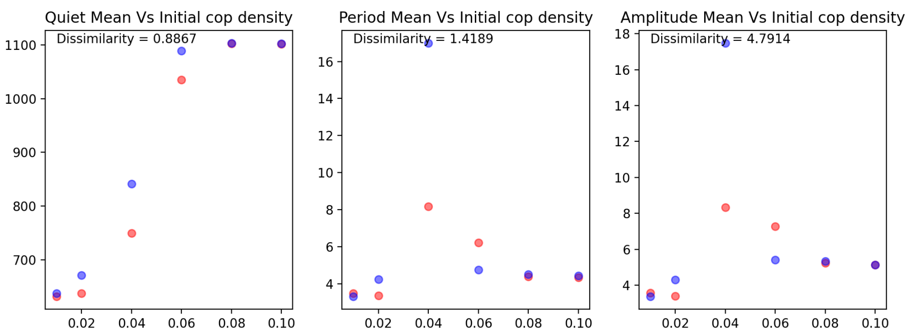

Figure 3: Cop Density Comparison

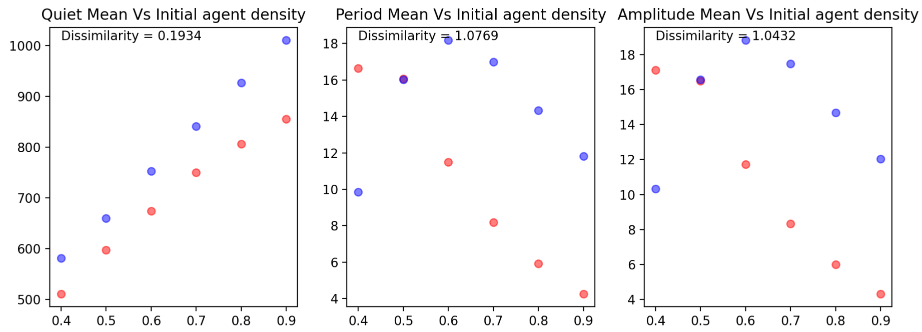

Figure 4: Agent Density Comparison

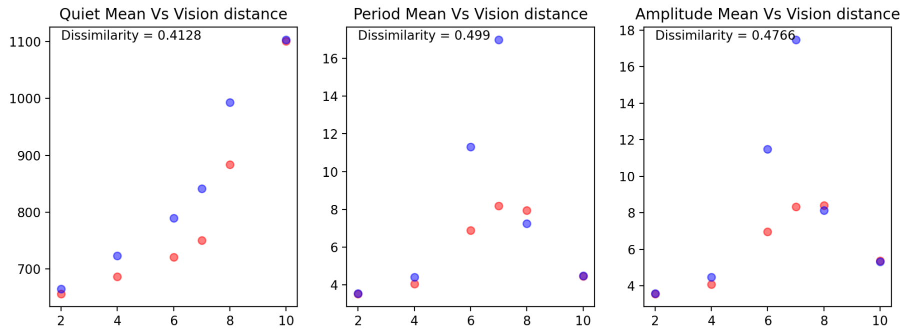

Figure 5: Vision Comparison

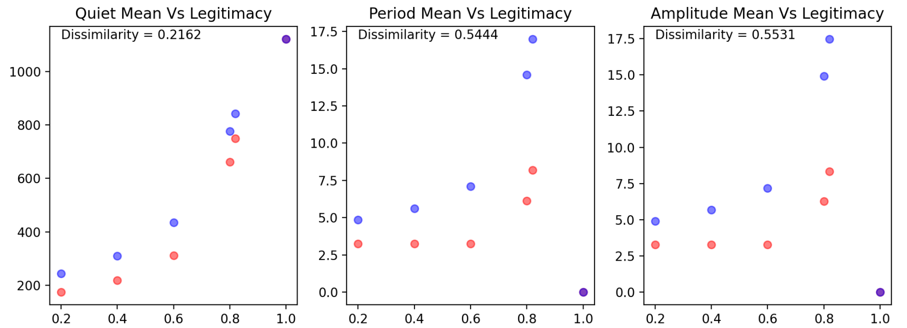

Figure 6: Government legitimacy Comparison

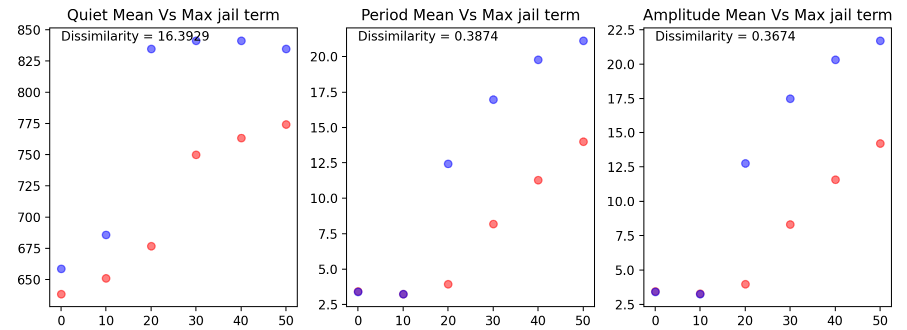

Figure 7: Max jail term Comparison

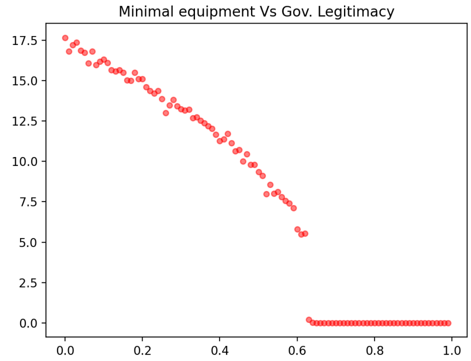

Figure 8: Minimal equipmentCoefficient Vs Gov. Legitmacy

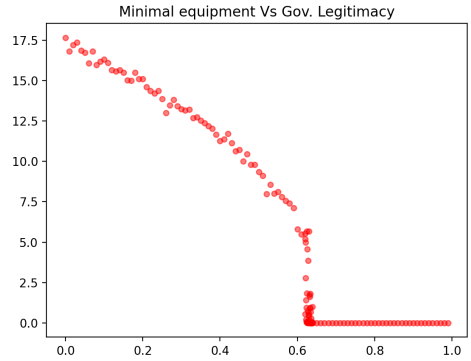

Figure 9: Minimal equipmentCoefficient Vs Gov. Legitmacy with extra data

4 Discussion

4.1 Model Comparison

We chose to measure these three values because they reflect how many people stay quiet generally, how often uprisings occur and how intense these uprisings are.

As you can see in Figure 2, the distributions of the average number of quiet agents, the average time between two uprisings and the average amplitude of two models have large overlapping areas.

We also used a statistical tool, KL divergence(Kullback & Leibler, 1951), to measure the distance between distributions, as an indicator of the dissimilarity between two models.

The large areas of overlap and the low value of the KL divergence(Kullback & Leibler, 1951) suggest that our replication is highly similar to the original model.

However, as Figure 2 clearly shows, for unknown reasons, our replication has a slightly higher central tendency. The impact of this difference is insignificant to the overall system, but it raised our attention and we made several speculations about it. One of which is that there could be small differences between the underlying low-level implementation of randomness in both models, or it could be that our sample size is simply too small. Whatever this reason is, it might also have caused some irregular dissimilarity values for the rest of the experiments' results.

For the rest of the experiments, we used following algorithm to measure the dissimilarity of how two models behave to different parameters:

$javaArray[sampleSize] \leftarrow javaData[sampleSize]$
$netlogoArray[sampleSize] \leftarrow netlogoData[sampleSize]$
assert $javaArray.length = netlogoArray.length$
$result \leftarrow 0$
$i \leftarrow 0$
while $i < javaArray.length - 1$ **do**
    $javaSlope \leftarrow javaArray[i+1] - javaArray[i]$
    $netlogoSlope \leftarrow netlogoArray[i+1] - netlogoArray[i]$
    $avgSlope \leftarrow avgSlope + abs((javaSlope - netlogoSlope)/netlogoSlope)$
    $i \leftarrow i + 1$
end while
return $avgSlope/javaArray.length$

It's clear to see that our model behaves similarly to the original model when facing changes in parameters.

The dissimilarity value remains low for the most experiments, but was surprisingly high for quiet mean Vs max-jail-term. Again, we are uncertain whether this is caused by possible differences in the implementation of randomness or simply an outlier due to the small size of sample space.

4.2 Research

After we conducted the experiment based on our initial design, we get results that are visualised into Figure 8.

It's clear to see when $L < 0.62$, our graph shows an linear relationship between these two variables, but afterwards there seems to be a huge gap in our plot.

We performed a linear regression analysis(Wikipedia, 2022) on the data points before 0.62, which generates:

$$y = -17.834x + 18.135$$
$$R^2 = 0.9661$$

We received a very high R^2, which indicates that these two variables are strongly correlated(Wikipedia, 2022) . Assuming the linear relationship remains after 0.62, when $L > 0.961$ even unequipped cops (`equipmentCoefficient` = 1) are able to preserve the stability. Nevertheless, the plot is clearly discontinuous and we need more information of the relationship around 0.62.

Therefore, in order to get a clear view on relationship between minimal `equipmentCoefficient` and L, we decided to run some extra experiments from 0.62 to 0.64. After filling in the extra data, we have Figure 9.

From Figure 9, it is clear to see that the equipment required for stability decreases as L increases and repidly reduces to zero after L reaches a certain threshold(0.62).

5 Conclusion

When $L \leq 0.62$, minimal `equipmentCoefficient` can be calculated by $y = -17.834x + 18.135$.

When $0.62 > L$, cops won't be needing equipments (simply `equipmentCoefficient` $= 1$ will suffice).

However, for this threshold(0.62), we are unsure if it is related to the rest of the parameters, such as initial cop density. To answer this question, further research based on our model is required.

6 Appendix

6.1 Group Work

Our group work division is almost the same as our plan in the proposal. Ho-Yu and Ruiming were doing the development of our replication and extension of the model. Zixuan was focusing on the model comparison and conducting experiments. One small difference is that every group member has been actively trying to help each other, so there's contribution from everyone in every task.

We've also been constantly communicating both online and in-person, having meeting on literally every major decisions. Even though there still are difficulties and disagreements along the way, we believe we've done an excellent job in regard to project management and teamwork.

6.2 Challenge

One of the challenges we confronted was time management. Because near the end of the semester, we all have a lot of assignments from other subjects. We completed each task later than we initially planned in the proposal. Fortunately, we shared our schedules and deadlines to find time slots that worked for all of us, and we managed to overcome this challenge through effective communication and collaboration.

Another challenge is not knowing how to compare the NetLogo model with our java implementation model at first. We studied some fundamental statistical tools to help us compare these two models.

References

Epstein, J. M. (2002). Modeling civil violence: An agent-based computational approach. *Proceedings of the National Academy of Sciences*, *99*(suppl 3), 7243–7250.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, *22*(1), 79–86.

Wikipedia. (2022). *Linear regression - wikipedia.* Retrieved May 26, 2022, from https://en.wikipedia.org/wiki/Linear_regression