Spring 2020

INFO6205 Final Project REPORT

COVID-19 SIMULATION

SHIHAO SUN, XULIANG MEI, JIASHU WANG

Introduction

In the past few months, the whole world has experienced an unprecedented disaster, the Covid-19 virus. So far (04/16), more than 2 million of people have been confirmed infectious over the world. The number of people who detected positive every day is still rising. In the US, many hospitals have become unrecognizable. Thousands of patients sick with the virus have swarmed into emergency rooms and intensive care units.

Many state governments declared civil emergency, closed schools and shops and called on citizens to keep social distance. But there is still a much grimmer set of questions: How many people are going to get hospitalized? How many of them will require critical care? When will they start showing up? Do we need to build extra hospitals to cure these people? Will we have enough ventilators?

There is no way to know those answers accurately. But we need to have an educated guess. That's the only way to be ready for a wave of Covid-19 patients. As the result, we conduct this project and try to simulate the spread of Covid-19 virus. Through the mathematical modeling of the spread of infectious diseases, we analyze the spread speed, spatial range, transmission route and kinetic mechanism of infectious diseases, in order to guide the effective prevention and control of infectious diseases.

Aim of the Project

Creating a mathematical model to simulate the spread of Covid-19 virus.

Providing relatively accurate data to guide the effective prevention and control of Covid-19.

Project Description

Basic assumptions of infectious disease model

As the first step in the modeling process, we identify the independent and dependent variables. The independent variable is time T, measured in days. And we use SEIR model as dependent variables, which includes 4 variables:

S = S(t) is the number of susceptible individuals;

E = E(t) is the number of exposed individuals;

I = I(t) is the number of infectious individuals;

R = (t) is the number of recovered individuals.

Susceptible person S first becomes exposed person E and goes through an incubation period, and then becomes infected person I after a period. Infected person I is found and becomes cured by treatment or by autoimmunity.

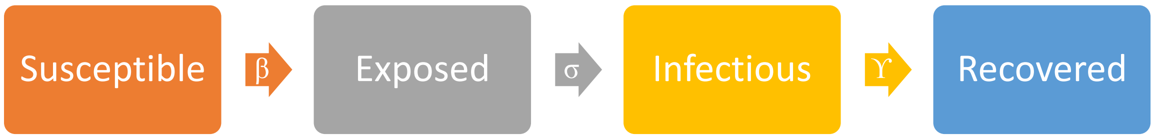


Figure 1

Suppose the total number of infected people in a region is N, if not isolated, each person will encounter r numbers of people every day, with β probability to infect these r people; exposed person E has a probability α of becoming infected; At the same time, the probability ϒ of an infected person becoming a convalescent R due to treatment or immunity. As a result, the functions can be set up as follows:

$$\frac{dS}{dt} = -\frac{\Upsilon\beta IS}{N}$$

$$\frac{dE}{dt} = \frac{\Upsilon\beta IS}{N} - \alpha E$$

$$\frac{dI}{dt} = \alpha E - \Upsilon I$$

$$\frac{dR}{dt} = \Upsilon I$$

$$\mathrm{N} = \mathrm{S} + \mathrm{E} + \mathrm{I} + \mathrm{R}$$

Optimized model for Covid-19

Actually, we now know that person who is in incubation period of Covid-19 virus could also be infective. As the result, we should take this situation into consideration. Assuming that the incubation will infect susceptible person S into a new exposed with probability β_2, and the number of people the exposed contacts every day is r_2, the model can be improved as following:

$$\frac{dS}{dt} = -\frac{\Upsilon\beta IS}{N} - \frac{r_2\beta_2 ES}{N}$$

$$\frac{dE}{dt} = \frac{\Upsilon\beta IS}{N} - \alpha E + \frac{r_2\beta_2 ES}{N}$$

$$\frac{dI}{dt} = \alpha E - \Upsilon I$$

$$\frac{dR}{dt} = \Upsilon I$$

$$\mathrm{N} = \mathrm{S} + \mathrm{E} + \mathrm{I} + \mathrm{R}$$

Implementation

Here, we use the normal distribution to simulate the population distribution and movement distribution.

This is the UML chart of our program:

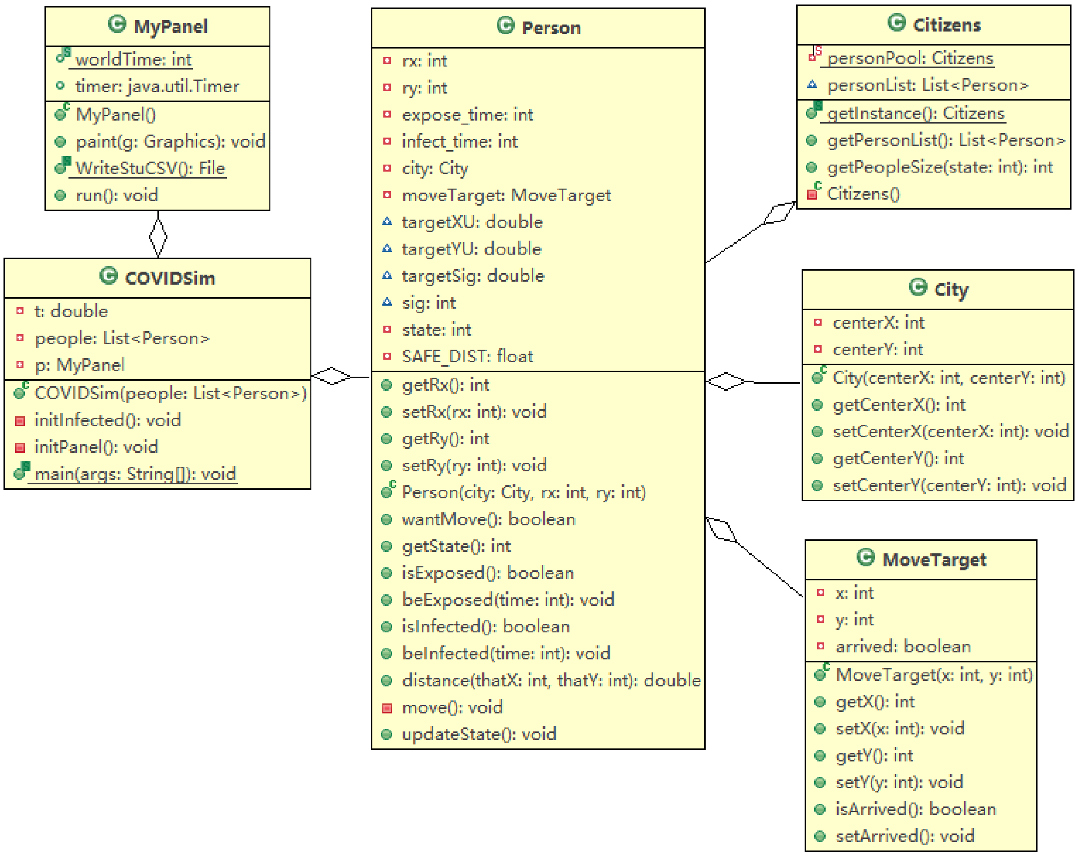


Figure 2 UML

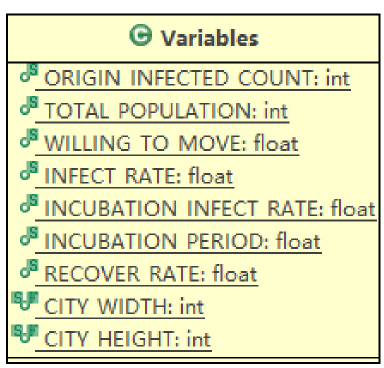


Figure 3 Variable.java

In Variables.java, we can change the variables such as the origin infected numbers, population, willing to move and so on to better simulate to real spread situation of the virus.

```
private void initInfected() {
    for (int i = 0; i < Variables.ORIGIN_INFECTED_COUNT; i++) {
        Person person;
        do {
            person = people.get(new Random().nextInt(people.size() - 1));// randomly pick a citizen
        } while (person.isInfected());// if the citizen is already infected, repick
        person.beInfected(0);// make this citizen to be infected
    }
}
```

This function is to initial the first group of infected people and start the spreading.

Then, person's state will be updated every day by updateState() function.

If the person is susceptible, it will check all the people within its safe distance, each people who is exposed or infect will have some possibility to make the person exposed.

```
if (state == State.SUSCEPTIBLE) {
    for (Person person : Citizens.getInstance().getPersonList()) {
        if(distance(person.getRx(), person.getRy()) > SAFE_DIST || person.getState() <= 1){
            continue;
        }
        float possibility = 0;
        if(person.getState() == 2) possibility = Variables.INCUBATION_INFECT_RATE;
        if(person.getState() == 3) possibility = Variables.INFECT_RATE;

        // if the random number is less than the exposed possibility, make this person to be exposed
        float random = new Random().nextFloat();
        if (random < possibility) {
            this.beExposed(MyPanel.worldTime);
            break;
        }
    }
}
```

If the person is already exposed, we use normal distribution to randomize the time he becomes infected.

```
if(state == State.EXPOSED){
    double randomInfectedTime = MathUtility.stdGaussian(25, Variables.INCUBATION_PERIOD / 2);
    if (MyPanel.worldTime - expose_time > randomInfectedTime && state == State.EXPOSED) {
        state = State.INFECTIOUS;// update state to be infectious
        infect_time = MyPanel.worldTime;// update time
    }
}
```

If the person is infectious, each day he will have RECOVER_RATE possibility to recover.

```
if(state == State.INFECTIOUS){
    float random = new Random().nextFloat();
    if(random < Variables.RECOVER_RATE){
        this.state = State.RECOVERED;
    }
}
```

If the person has recovered, there is no extra process.

Output

In our program, we set up a city which width is 700 and length is 800. There are 5000 people in this city. Figure 4 is the user interface of our simulation program.

The black point is the susceptible person, who is health right now but has the possibility to be infected.

The yellow point is the exposed and incubated person.

The red point is infectious person.

The green point is the person who has recovered.

In our program, we set the origin infected people number as 50. The infect rate β is 0.8, incubation infect rate is 0.5, recover rate is 0.01. These values are set based on the research of Covid-19. And we can change the WILLING_TO_MOVE value to simulate different epidemic prevention policy. 0.99 is totally uncontrolled situation, everyone moves frequently without any prevention measures. -0.99 means everyone stays at home, nobody is moving.

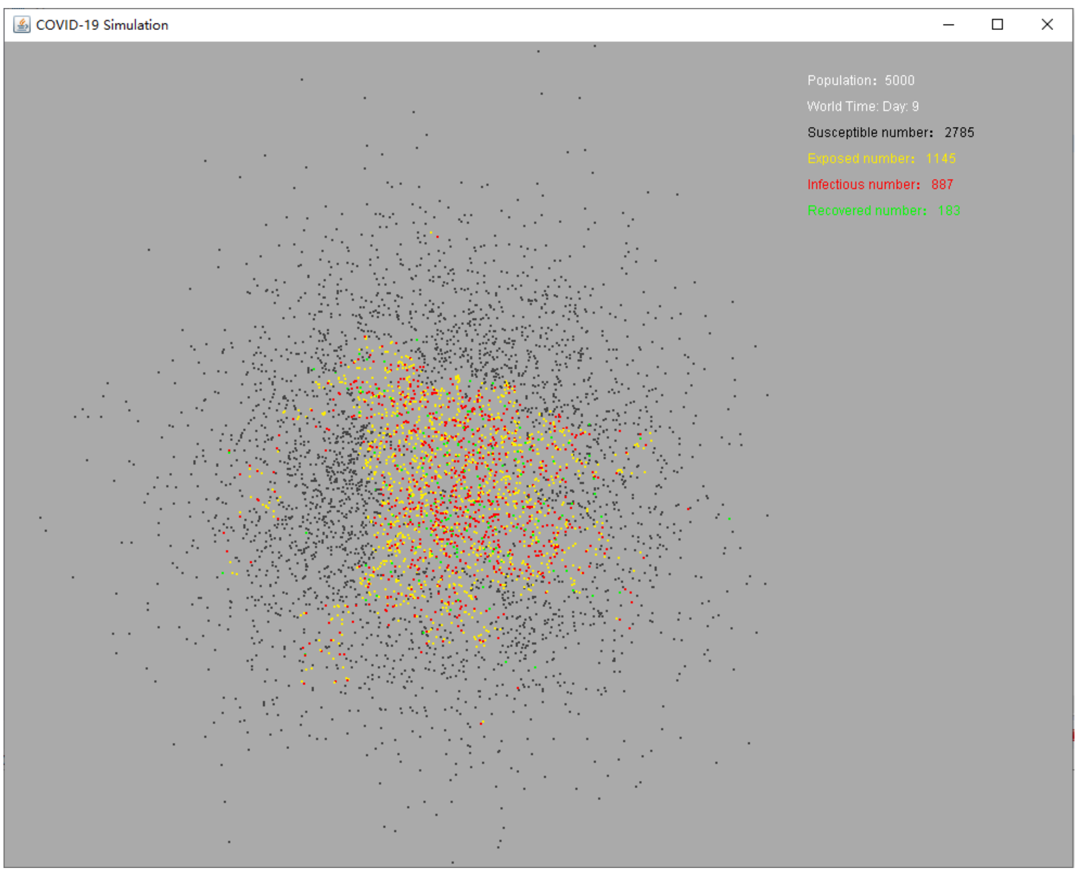


Figure 4 Simulation Interface

History Git Staging Problems Console

```
<terminated> COVIDSim [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe
======= DAY 1.0 ========
Susceptible number: 4884
Exposed number: 65
Infectious number: 48
Recovered number: 3
======= DAY 2.0 ========
Susceptible number: 4814
Exposed number: 131
Infectious number: 49
Recovered number: 6
======= DAY 3.0 ========
Susceptible number: 4700
Exposed number: 232
Infectious number: 58
Recovered number: 10
```

Figure 5 Console Output

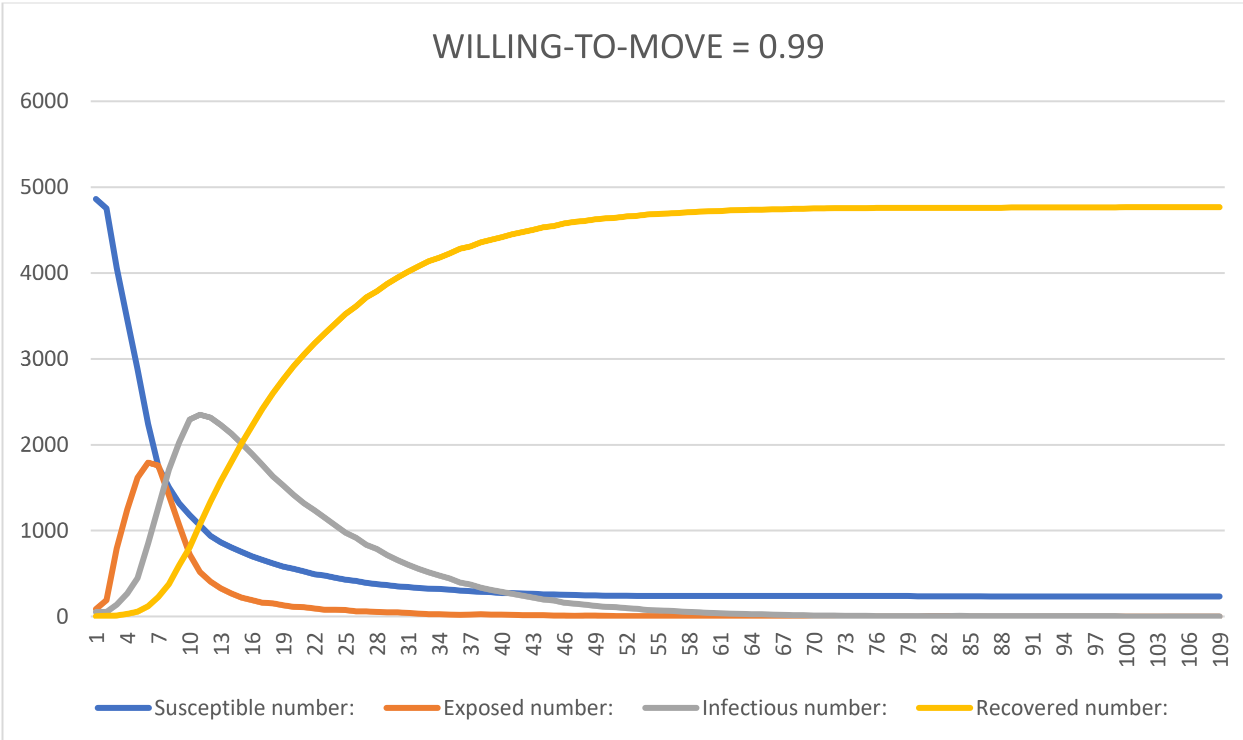
	A	B
1		
2	======= DAY 1.0 =======	1
3	Susceptible number:	4862
4	Exposed number:	86
5	Infectious number:	50
6	Recovered number:	2
7	======= DAY 2.0 =======	2
8	Susceptible number:	4753
9	Exposed number:	188
10	Infectious number:	52
11	Recovered number:	7

Figure 6 CSV File

Experiment 1

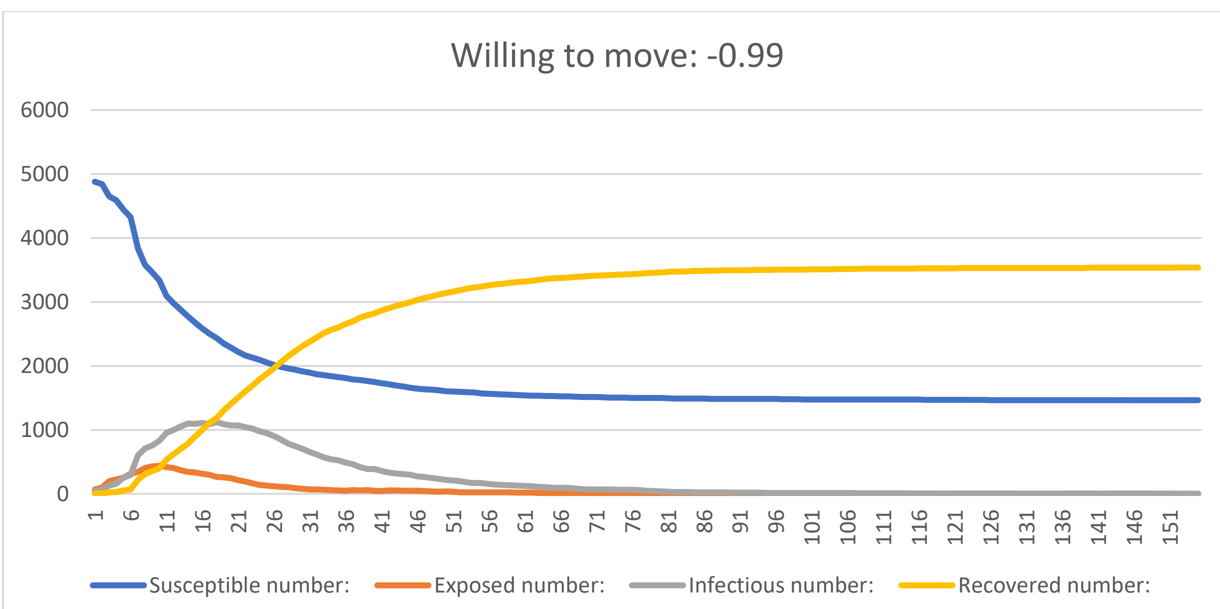
First, WILL-TO-MOVE is 0.99, the infected people are not quarantined, and everyone keeps moving. And the INFECT-RATE is 0.8, INCUBATION_INFECT_RATE is 0.5. That means people do not wear face masks to protect themselves and they are easy to be infected.

With time passing by, the exposed population peak, then reach an inflection point as the exposed become infected, and as treatment and autoimmunity reduce the number of exposed and infections and increase the number of recovered people. At about day 11, the infected population reach the top point, which is about 2500 people. And it takes more than 100 days for the virus to disappear in this small city.



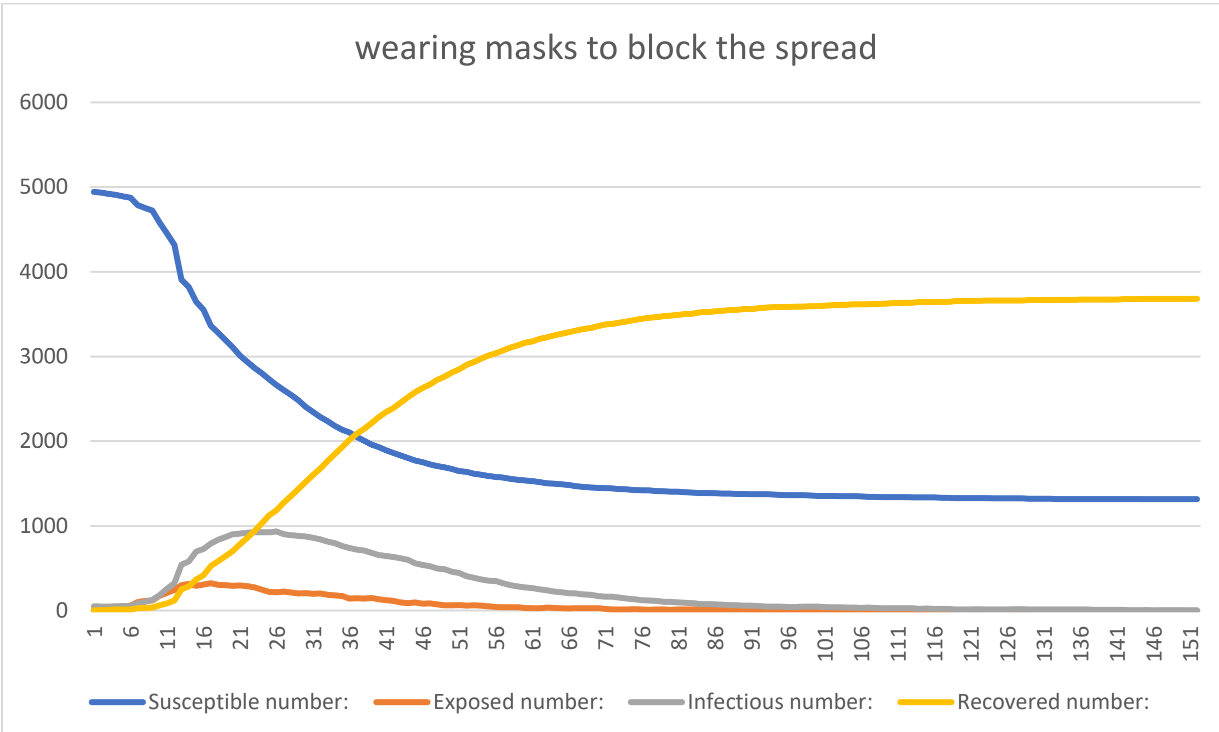
Experiment 2

Then, we set WILLING-TO-MOVE as -0.99. Other variables are unchanged. This time, less people would be infected, which is only about 1000 people. And it takes about 16 days to reach the peak. That means the government and hospital would have more time to be prepared. And in the end, the total number of infected people (including exposed and recovered) is about 3500, which is 30% less then experiment 1. Under this circumstance, the hospital will have less pressure and need less equipment.



Experiment 3

This time, we change the rate of infection. If all citizens wear face masks when moving, the rate of infection could be reduced to 1/10 of the normal situation. So, we set INFECT-RATE as 0.08, INCUBATION_INFECT_RATE as 0.05. WILLING-TO-MOVE is still 0.99. As we can see in the chart, compared with experiment 1, less people are infected, and the peak of infection is much lower.



Conclusion

In this project, we use SEIR model to simulate the spread of Covid-19 virus. Based on SEIR model, we add incubation infective into consideration to improve our model. After some simulation experiments, we can conclude that, if we keep staying at home and maintaining social distance with each other, or wear face masks, we can effectively reduce the spread of the epidemic and the number of infected people. If we do nothing and just live as normal, finally most of the citizens would be infected.