UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

MASTER ICT

MI2.05 – MODELLING AND SIMULATION OF COMPLEX SYSTEMS

LE Huy Duc

May 2020

1 Overview

- In this project, we use the shape files of Phuc Xa for buildings and road.

- We program a global time counter, which allows us to monitor the current date and current time of the day.

- Values of epidemiological parameters are hard-coded constants.

- We create a species "human" that implements all human activities. It contains gender, age, where and when to go, …

- We use colors to denote different stage of SEIR: **Susceptible**, **Exposed**, **Infected**, **Recovered**.

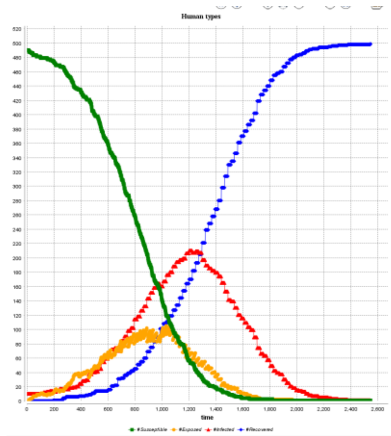
- Finally, the results of exploration models are quite long. So we leave the result in the appendix at the end of this document. We provide explanations for the results in the Appendix.

- To see the full model, please read E4.gaml

2 Session 1 - Spread of the disease

2.1 Model M1-1

This model successfully implements SEIR model and plot the number of people in each SEIR stage.



2.2 Model M1-2

- Adding R0 is very simple. Each human as an attribute *"myR0"* and action *"infect_others"*. Whenever a human successfully infect another human, we increase his "*myR0*" by 1.

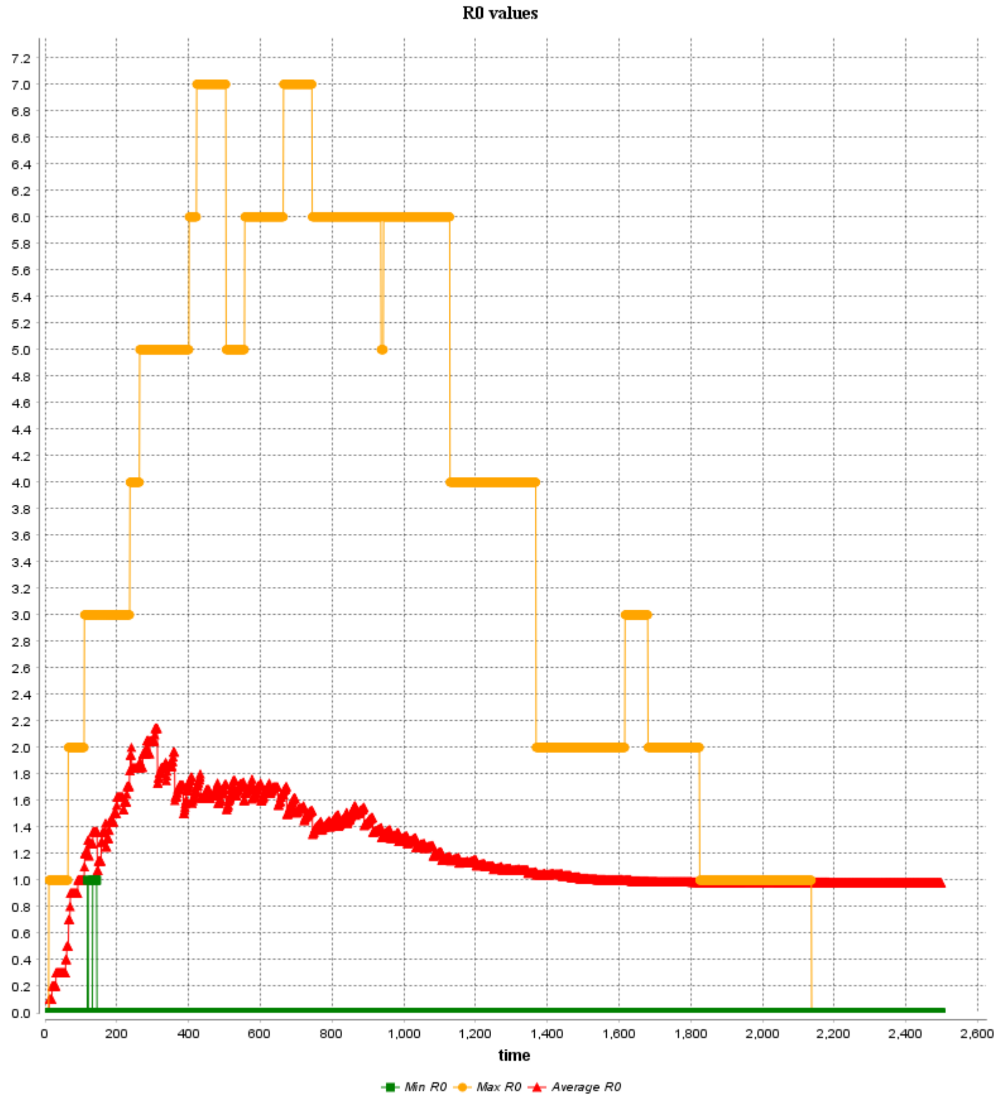
```
reflex infect_others when: isInfected = true {
    float prob <- myTransmissionProb;

    list<human> close_humans <- (human at_distance(distanceContamination)) where(each.isSusceptible = true);
    loop man over: close_humans {
        if flip(prob) {
            myR0 <- myR0 + 1;
            ask man {
                do set_exposed();
            }
        }
    }
}
```

- Finding average R0 can be done with:

```
(human sum_of(each.myR0)) / max(1, nbRecovered + nbInfected)
```

- Comments:

  + The R0 is high at the start, but it slowly decreases as there are fewer people to infect.



- We update the status of each human agent daily. This is why there are many discrete steps instead of a continuous line.

2.3 Explorations E1-1

- For this task, we use the statement:

```
init {
    create simulation with:[humanNumber::humanNumber, initialInfected::initialInfected, seed::2];
    create simulation with:[humanNumber::humanNumber, initialInfected::initialInfected, seed::3];
```

to create simulations and set seed.

- The result is in the appendix.

2.4 Explorations E1-2

```
parameter "Number of people" var: humanNumber init: 200 min:200 max:2000 step: 200;
parameter "Initial infected" var: initialInfected init:10 min:1 max:100;

init {
    create simulation with:[humanNumber::400, initialInfected::initialInfected, seed::seed];
    create simulation with:[humanNumber::600, initialInfected::initialInfected, seed::seed];
    create simulation with:[humanNumber::800, initialInfected::initialInfected, seed::seed];
    create simulation with:[humanNumber::1000, initialInfected::initialInfected, seed::seed];
```

- We use the same graphs as in Model M1-1.

2.5 Explorations E1-3

- Finding the cycle with max number of infected is easy. Whenever someone become infected, we do:

```
if nbInfected > maxInfectedCount {
    dayWithMaxInfected <- currentDay;
    cycleWithMaxInfected <- cycle;
    maxInfectedCount <- nbInfected;
}
```

- The .csv file is sent along with the report.

3 Session 2 - Spread in a city with heterogeneous population

3.1 Model M2-1

*** Note:** M2-1 is merged into file M2-2.gaml. Also, we do not color the buildings because it's harder to see infected *human* agents.

```
// ENUMS for locations
int NB_TYPE <- 9;
string TP_HOME <- "Home";
string TP_INDUSTRY <- "Industry";
string TP_OFFICE <- "Office";
string TP_SCHOOL <- "School";
string TP_SHOP <- "Shop";
string TP_SUPERMARKET <- "Supermarket";
string TP_CAFE <- "Cafe";
string TP_RESTAURANT <- "Restaurant";
string TP_PARK <- "Park";
list<string> BUILDING_TYPES <- [TP_HOME, TP_INDUSTRY, TP_OFFICE, TP_SCHOOL, TP_SHOP, TP_SUPERMARKET, TP_CAFE, TP_RESTAURANT, TP_PARK];
```

- We have 9 types of buildings. We use enums instead of raw string because using hard-coded strings are bad practice.

- We ensure that there are at least one of each building and two schools. And the number of house-type building is around 85%.

```
loop i from: 0 to: NB_TYPE - 1{
    building[i].buildingType <- BUILDING_TYPES[i];
}

building[NB_TYPE].buildingType <- TP_SCHOOL;

loop i from:NB_TYPE+1 to: length(building)-1 {
    if flip(0.85) {
        building[i].buildingType <- TP_HOME;
    }
    else {
        building[i].buildingType <- one_of(BUILDING_TYPES);
    }
}
```

3.2 Model M2-2

- In this model, we initialize the home and workplace for each human entity in:

```
create human number: humanNumber {
    myHouse <- one_of(building where (each.buildingType = TP_HOME));
    myWorkplace <- one_of(building where (each.buildingType != TP_HOME));   // 1 is home. Other is not home
```

- We make a generalize *gotoLocation()* action that tells a *human* to go somewhere. Using that, we can create many behaviors for different types of human (child, adult, …) in the next model.

```
action gotoLocation(building targetPlace) {
    bool arrived <- first(building overlapping(self)) = targetPlace;

    if (!arrived) {
        if (contagionInTransport) {
            do goto target: any_location_in(targetPlace) on: roadNetwork speed: (rnd(100)+1)    #m/#s;
        }
        else {
            location <- any_location_in(targetPlace);
        }
    }
}

reflex goWorkOrHome {
    bool isAtWork <- first(building overlapping(self)) = myWorkplace;
    bool isAtHome <- first(building overlapping(self)) = myHouse;

    if (currentHour >= startWork and currentHour <= endWork) {
        if (!isAtWork) {
            do gotoLocation(myWorkplace);
        }
    }
    else {
        if(!isAtHome) {
            do gotoLocation(myHouse);
        }
    }
}
```

- Finally, in this model, we update the way people infect others. Each person tries to infect all people in the building one time.

```
reflex infect_others when: isInfected = true {
    float prob <- myTransmissionProb;

    building myBuilding <- first(building overlapping(self));
    list<human> close_humans <- (human overlapping(myBuilding)) where(each.isSusceptible = true);
    loop man over: close_humans {
        if flip(prob) {
            myR0 <- myR0 + 1;
            ask man {
                do set_exposed();
            }
        }
    }
}
```

3.3 Model M2-3

- We use 4 types of humans: child (age <= 3), students (age 4-22), adults (age 23-55) and retiree (age > 55).

```
create human number: humanNumber {
    myHouse <- one_of(building where (each.buildingType = TP_HOME));
    myWorkplace <- one_of(building where (each.buildingType != TP_HOME));   // 1 is home. Other is not home
    mySchool <- one_of(building where (each.buildingType = TP_SCHOOL));
    location <- myHouse.location;

    gender <- one_of(["male","female"]);
    age <- rnd(70) + 1;

    if nbInfected < initialInfected {
        do set_infected();
    }
}
```

- Each type has a different agenda:
  + Children will do nothing.
  + Students will go to school during school hour
  + Adults go to work.
  + Old people go to random places.

- In Model M3, we will add a new activity: go to buy food. This activity can happen even in lockdown.

*** Note:** we do not implement the part that requires 1 household per building because that will lead to too many people.

```
reflex go_child when: age<=3 {
    // do nothing
}

reflex go_student when: age >3 and age<=22 {
    bool isAtSchool <- first(building overlapping(self)) = mySchool;
    bool isAtHome <- first(building overlapping(self)) = myHouse;

    if (currentHour >= startSchool and currentHour <= endSchool) {
        if (!isAtSchool) {
            do gotoLocation(mySchool);
        }
    }
    else {
        if(!isAtHome) {
            do gotoLocation(myHouse);
        }
    }
}

reflex go_adult when: age>22 and age<=55 {

    bool isAtWork <- first(building overlapping(self)) = myWorkplace;
    bool isAtHome <- first(building overlapping(self)) = myHouse;

    if (currentHour >= startWork and currentHour <= endWork) {
        if (!isAtWork) {
            do gotoLocation(myWorkplace);
        }
    }
    else {
        if(!isAtHome) {
            do gotoLocation(myHouse);
        }
    }
}

reflex goRetiree when: age > 55 {
    // do nothing
    if (currentHour <= endWork) {
        building targetLocation <- one_of(building);
        do gotoLocation(targetLocation);
    }
    else {
        do gotoLocation(myHouse);
    }
}
```

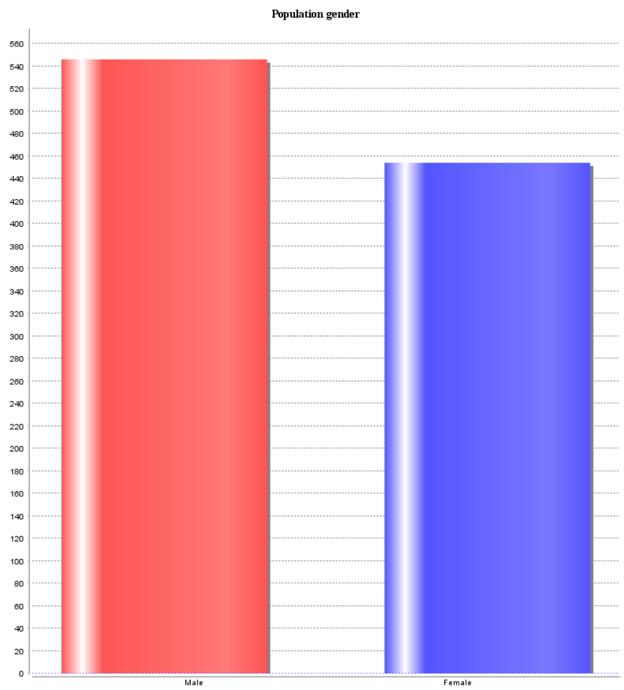
Figure 1. Agendas of human agents

3.4 Exploration E2-1

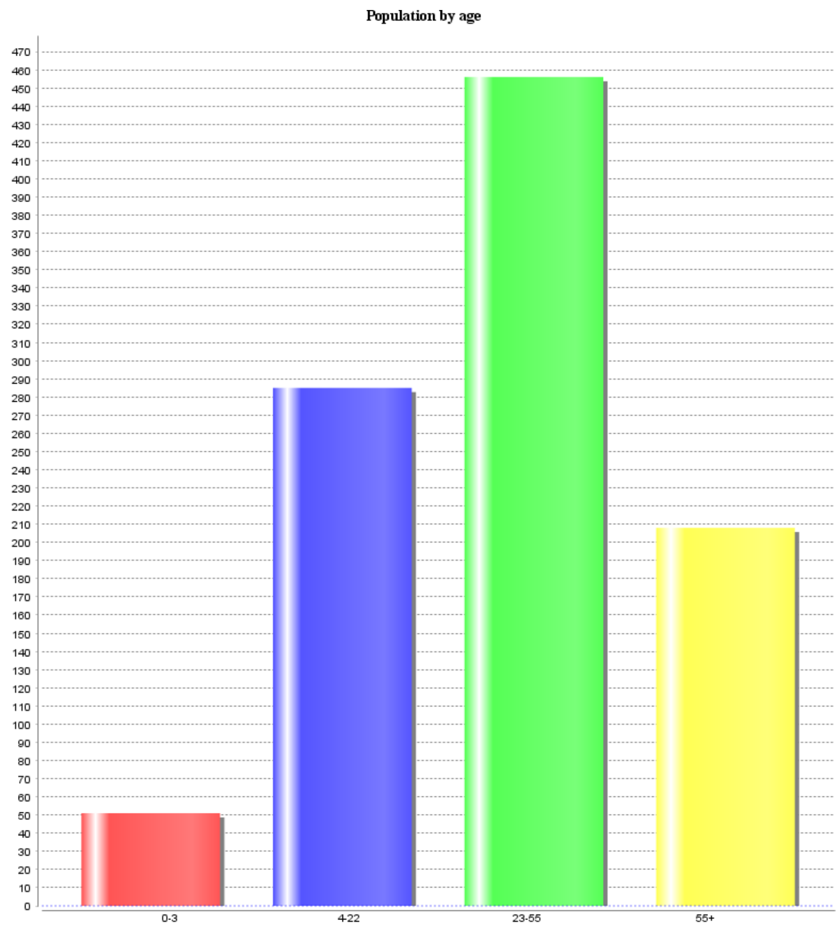
- See appendix.

3.5 Exploration E2-2

- Graph 1 is population by gender. We see that there are more **Male** than **Female**. However, this value is completely random and can change depending on the seed of the experiment.



- Graph 2 is population by age. We ensure that majority of the population are either students or adults. The average age is 36.



3.6 Exploration E2-3, E2-4

- For E2-3, we've already plot everything since model 1-1. So it's already done.

- To add environmental transmission, we add some attributes and reflex to buildings.

```
string buildingType;
bool isInfected <- false;
float infectionLevel <- 0;
int lastDay <- 0;

reflex update_infection_level {
    if contagionInBuilding = false{
        return;
    }
    list<human> nearInfectedHumans <- (human overlapping(self))  where (each.isInfected);
    infectionLevel <- infectionLevel + transmissionProb * 0.1 * length(nearInfectedHumans);

    if (lastDay != currentDay) {
        infectionLevel <- max(0,infectionLevel*0.5 - 0.1);
        lastDay <- currentDay;
    }
}
```

*** Note**: there are 2 statements that need clarification:

a) `infectionLevel <-infectionLevel+transmissionProb * 0.1 * length(nearInfectedHumans);`

+ We multiply by 0.1 because the probability of environmental transmission should be much smaller than direct human transmission.

b) `infectionLevel <- max(0,infectionLevel*0.5 - 0.1);`
+ Each day, the infection level is reduced by more than half (exponential decay).

- Finally, we add actions for *human* species to simulate getting infected from building.

```
reflex exposed_from_environment {
    if isSusceptible = false or contagionInBuilding = false{
        return;
    }

    building current_building <- first(building overlapping(self));
    if current_building = nil {
        return;
    }
    if (flip(current_building.infectionLevel)) {
        do set_exposed();
    }
}
```

- For the result of Exploration 2-4, see Appendix.

4 Session 3 - Public health policy

4.1 Model M3-1

- We merge model 3-1 and 3-2 into one. So there's only M3-2.gaml file.
- For this model, we define the species LocalAuthority. It has 3 main tasks.

a) Control when to apply policy

```
species LocalAuthority {
    int lastDay <- 0;

    bool shouldStartPolicy {
        //write("should we apply");
        if (policyTriggerMode = POLICY_BY_TIME) {
            if currentDay >= timeToApply {
                return true;
            }
            else {
                return false;
            }
        }

        if (policyTriggerMode = POLICY_BY_INFECTED) {
            if nbConfirmed >= infectedToApply {
                return true;
            }
            else {
                return false;
            }
        }
    }
```

b) What happens when a policy is applied (close buildings, enforce wearing mask, …)

```
reflex applyPolicy when: shouldStartPolicy() {
    //write("applying policies");
    if policyNoSchool > 0 {
        isAllowed[TP_SCHOOL] <- false;
    }

    if policyTotalLockdown > 0 {
        loop i from:0 to: NB_TYPE - 1{
            isAllowed[BUILDING_TYPES[i]] <- false;
        }
    }

    isAllowed[TP_SUPERMARKET] <- true;
    //******************************
    if policyWearMask > 0 {
        requireMask <- true;
    }

    if policyContainByAge > 0 {
        requireContainByAge <- true;
    }
}
```

- *isAllowed[buildingType]* defines whether a building is open or not. For example, if *isAllowed[TP_SCHOOL]=false,* it means schools are closed.

c) Testing random people. Our model also takes into consideration false positive and false negative:

```
bool hasDisease(human person) {
    return (person.isExposed or person.isInfected);
}

reflex randomTest when: shouldStartPolicy() and currentDay!=lastDay { // do test 1 times at the beginning of the day
    lastDay <- currentDay;
    write("testing");

    // test random unconfirmed people to see if they're postive.
    list<human> testedHumans <- nbTestPerDay among (human where(each.isConfirmed=false));

    loop person over: testedHumans {
        if (hasDisease(person) and flip(truePositiveRate)) or (!hasDisease(person) and flip(1-trueNegativeRate)) {
            person.isConfirmed <- true;
            nbConfirmed <- nbConfirmed + 1;
            ask person {
                do startQuarantine();
            }
        }
    }
}
```

- When someone is “confirmed” (whether true positive or not), he is forced to stay at home. This applies to all ages.

```
reflex go_adult when: age>22 and age<=55 {
    if (not isAllowed[myWorkplace.buildingType] or isConfirmed) {
        return;
    }
```

4.2 Model M3-2

- When a human becomes infected using the action “set_infected()”, there’s a probability that he’s asymptomatic.

```
action set_infected {
    isSusceptible <- false;
    isExposed <- false;
    isInfected <- true;
    isRecovered <- false;

    color <- #red;

    infectiousDay <- currentDay;
    nbExposed <- nbExposed - 1;
    nbInfected <- nbInfected + 1;

    asymptom <- flip(asymptomProb);
}
```

- People who are asymptomatic are less likely to infect others. Wearing mask also give the same effect.

```
reflex infect_others when: isInfected = true {
    float prob <- transmissionProb;
    if requireMask {
        prob <- prob * 0.5;
    }
    if asymptom {
        prob <- prob * asymptomReduction;
    }
```

4.3 Exploration E3-1

- see Appendix.

5 Session 4 - Space of freedom

5.1 Realistic movement

- We use the variable “bool contagionInTranports;” to choose method of movement.

- Value “false” means that people arrive at their destination immediately. “True” means people have to use the road networks, which means there’re more chance of getting infected during commute. We use this in E4. This affects the way people are infected.

```
reflex infect_others when: isInfected = true {
    float prob <- transmissionProb;
    if requireMask {
        prob <- prob * 0.5;
    }
    if asymptom {
        prob <- prob * asymptomReduction;
    }

    building myBuilding <- first(building overlapping(self));
    if myBuilding = nil and contagionInTransport {  // if not in building, then use contagion in transport
        list<human> close_humans <- (human at_distance(distanceContamination)) where(each.isSusceptible = true);
        loop man over: close_humans {   // contact each person 1 time
            if flip(prob) {
                myR0 <- myR0 + 1;
                ask man {
                    do set_exposed();
                }
            }
        }
    }
    else {  // else use rules for contagion in building
        list<human> close_humans <- (human overlapping(myBuilding)) where(each.isSusceptible = true);
        loop man over: close_humans {   // contact each person 1 time
            if flip(prob) {
                myR0 <- myR0 + 1;
                ask man {
                    do set_exposed();
                }
            }
        }
    }
}
```

5.2 Free food policy

- In model M3-2, the policies close many buildings. But Supermarkets is always open so that people can go to buy good.

- But this means people can get infected from going to the supermarket or at the supermarket.

=> This policy means food is delivered to everyone at home, which means they don’t have to go to supermarkets => reduce chance of getting infected.

- We test this policy in E3 and E4. The result is in the Appendix.

```
reflex goBuyFood when: age > 10 and lastBuyFoodDay!=currentDay and (flip(0.3) or goingToSupermarket or goingBackHome) {
    if isConfirmed or policyFreeFood > 0
       or (age>3 and age<=22 and currentHour>startSchool and currentHour<endSchool and isAllowed[TP_SCHOOL])
       or (age>22 and age<=55 and currentHour>startWork and currentHour<endWork and isAllowed[myWorkplace.buildingType])
    {
        return;
    }


    if goingToSupermarket = false and goingBackHome = false {
        goingToSupermarket <- true;
        targetSupermarket <- one_of(building where(each.buildingType = TP_SUPERMARKET));
    }

    building currentBuilding <- first(building overlapping(self));
    if goingToSupermarket {
        //write("Buying food");
        do gotoLocation(targetSupermarket);

        if currentBuilding = targetSupermarket { // arrive at the supermarket and buy food successful
            //write("reached market");
            lastBuyFoodDay <- currentDay;
            goingToSupermarket <- false;
            goingBackHome <- true;
        }
    }

    if goingBackHome {
        //write("Going home");
        do gotoLocation(myHouse);

        if currentBuilding = myHouse {
            goingBackHome <- false;
        }
    }
}
```

Figure 2. Agenda: going to supermarket

6 Conclusion/Summary

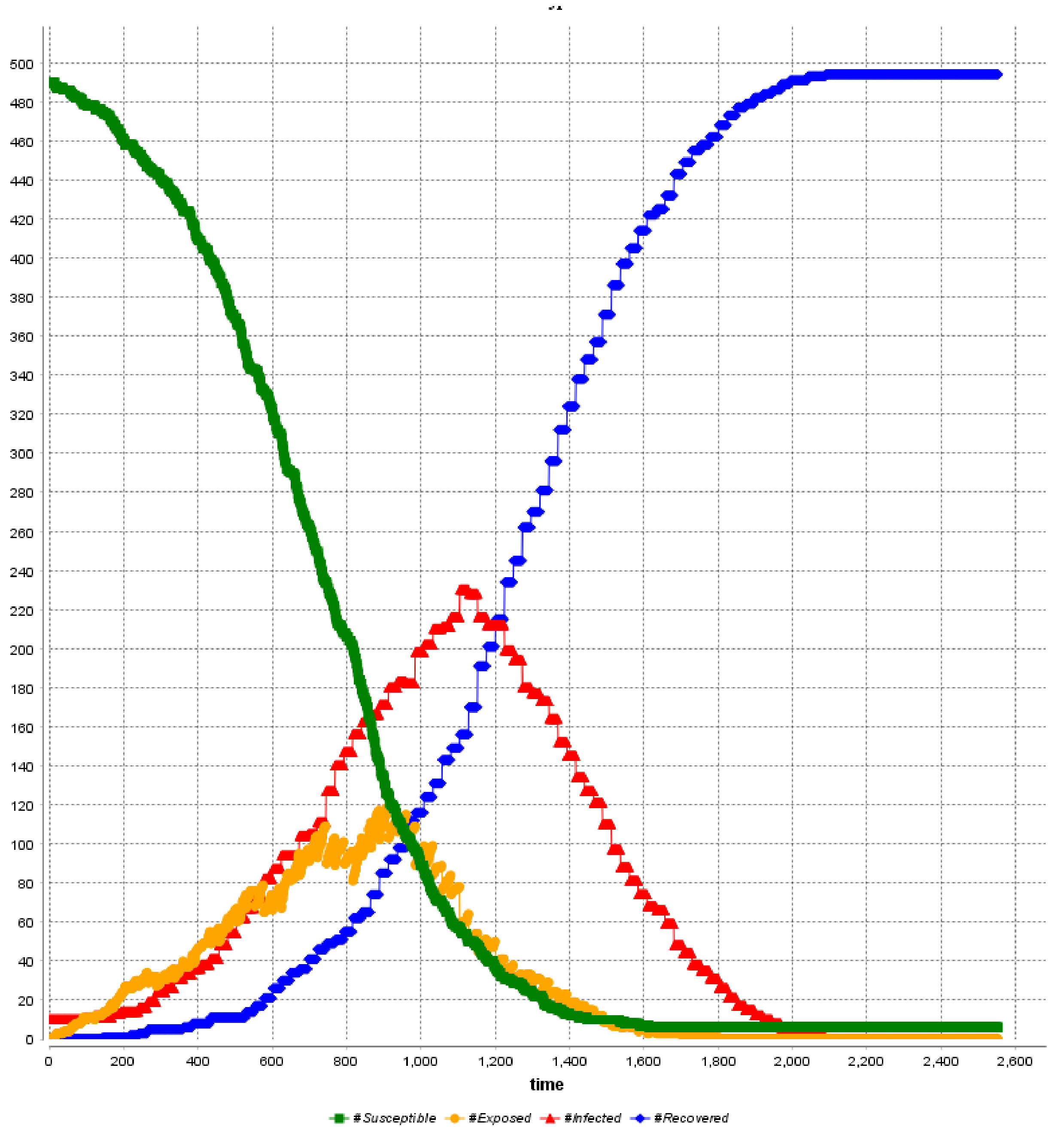
In this project, we have implemented all required features as well as explaining the results. Also, we've added the ability to simulate contagion in transport and the Free-food Policy.

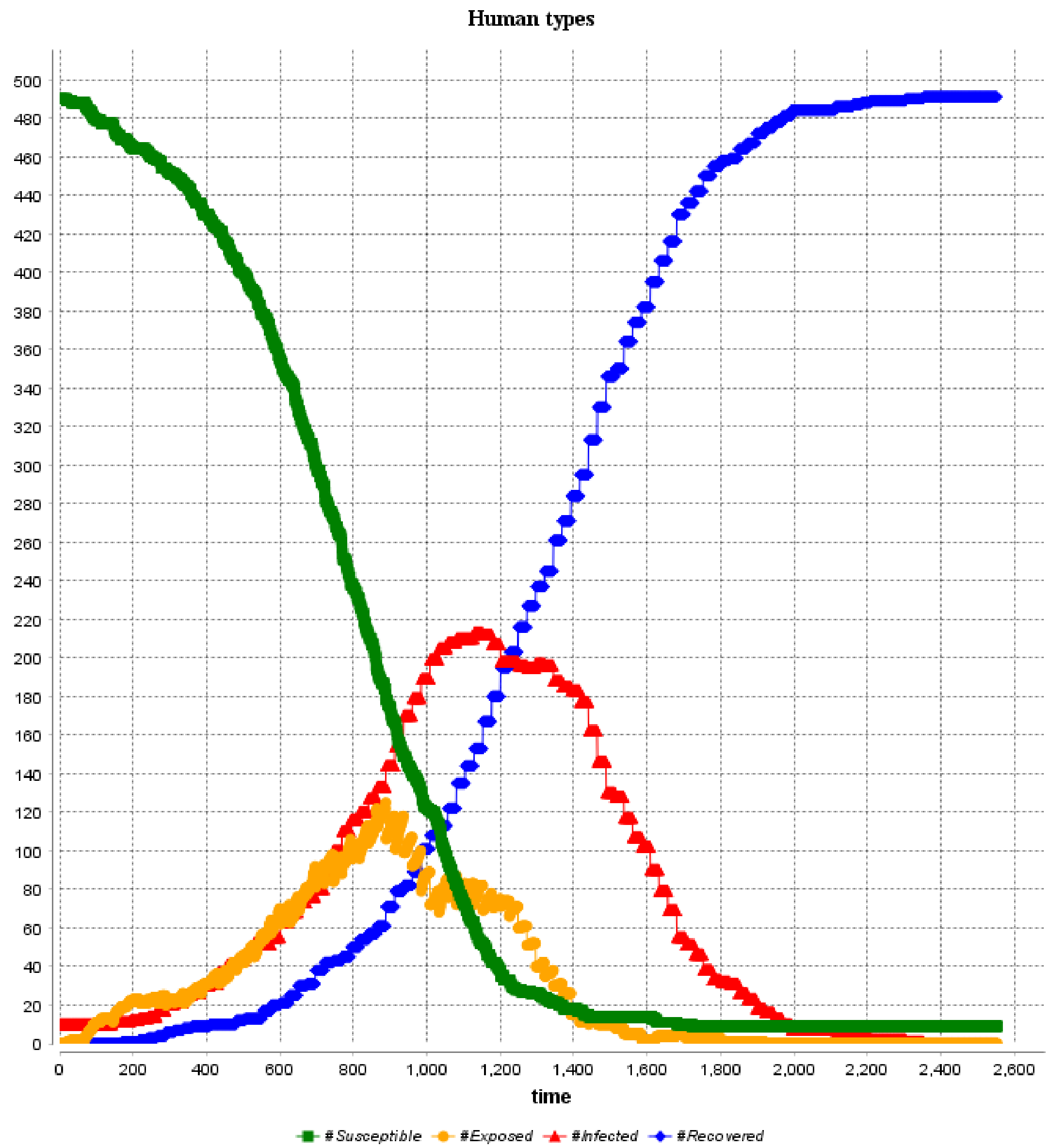
For the full model with everything (including code comment) see file E4.gaml

A Appendix

A.1 E1-1

- There are small different due to randomness, but in both graphs, the number of infected is bell-shaped.



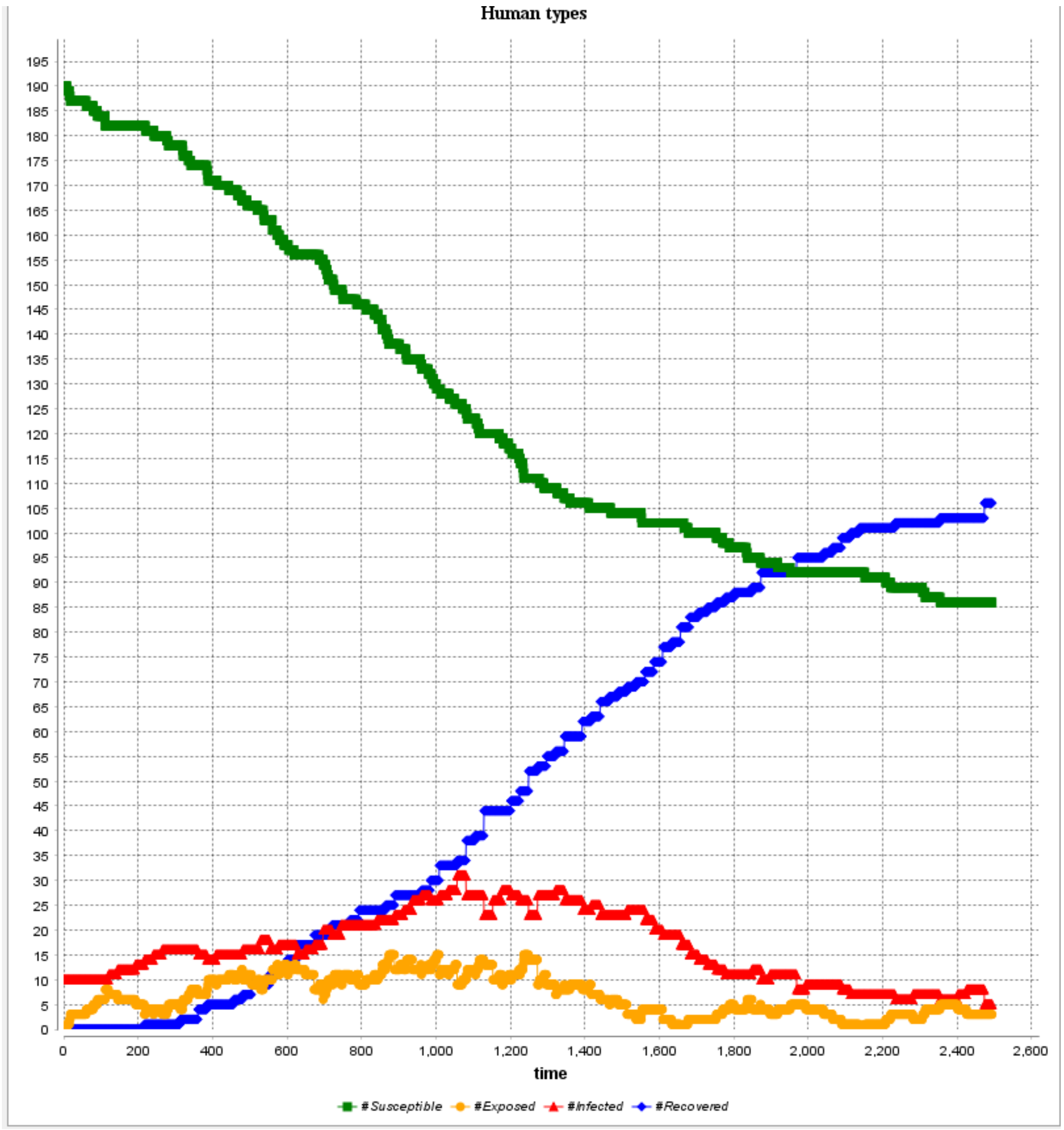

Human types
500
480
460
440
420
400
380
360
340
320
300
280
260
240
220
200
180
160
140
120
100
80
60
40
20
0
0
200
400
600
800
1,000
1,200
1,400
1,600
1,800
2,000
2,200
2,400
2,600
time
#Susceptible
#Exposed
#Infected
#Recovered

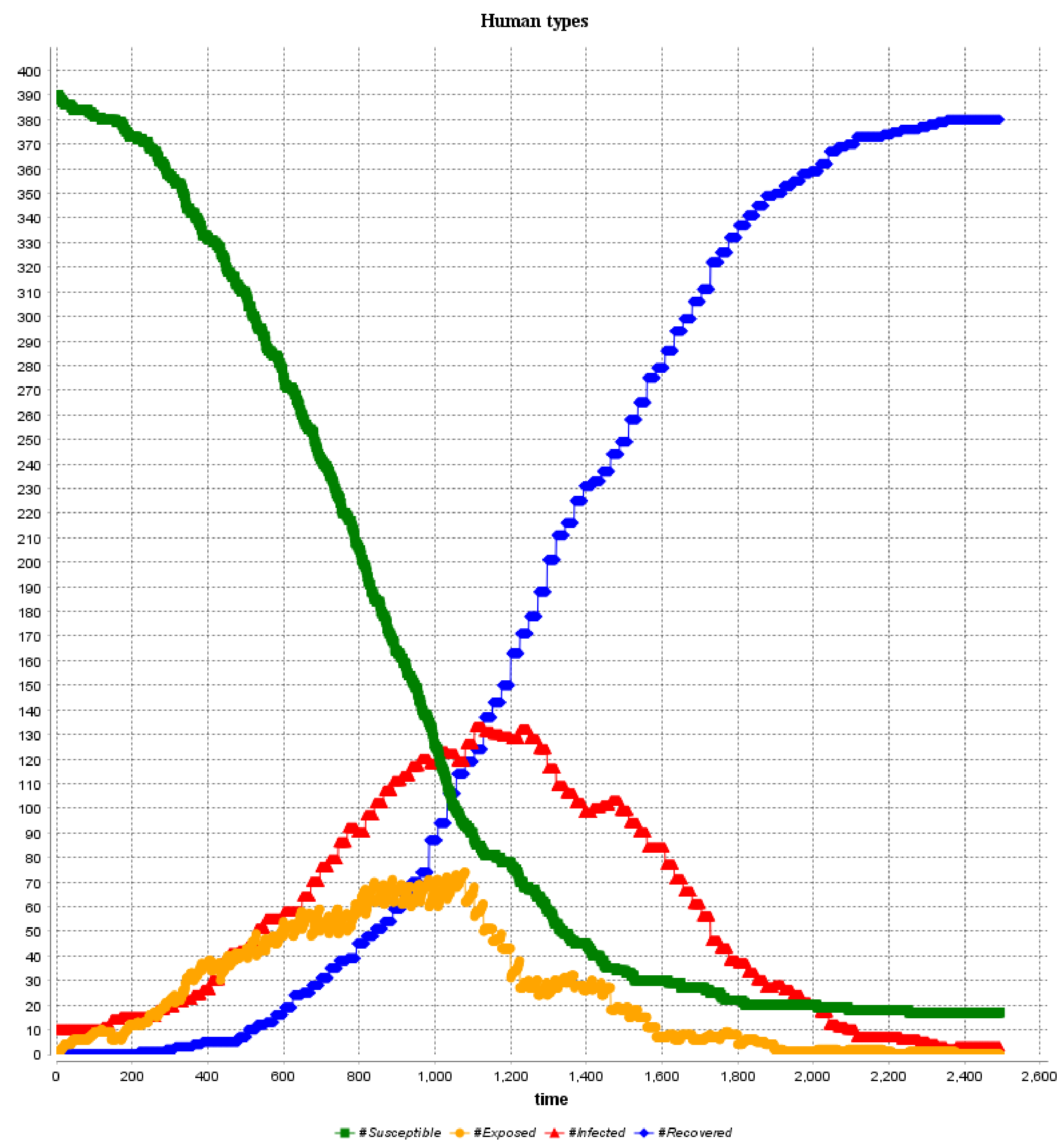
A.2 E1-2

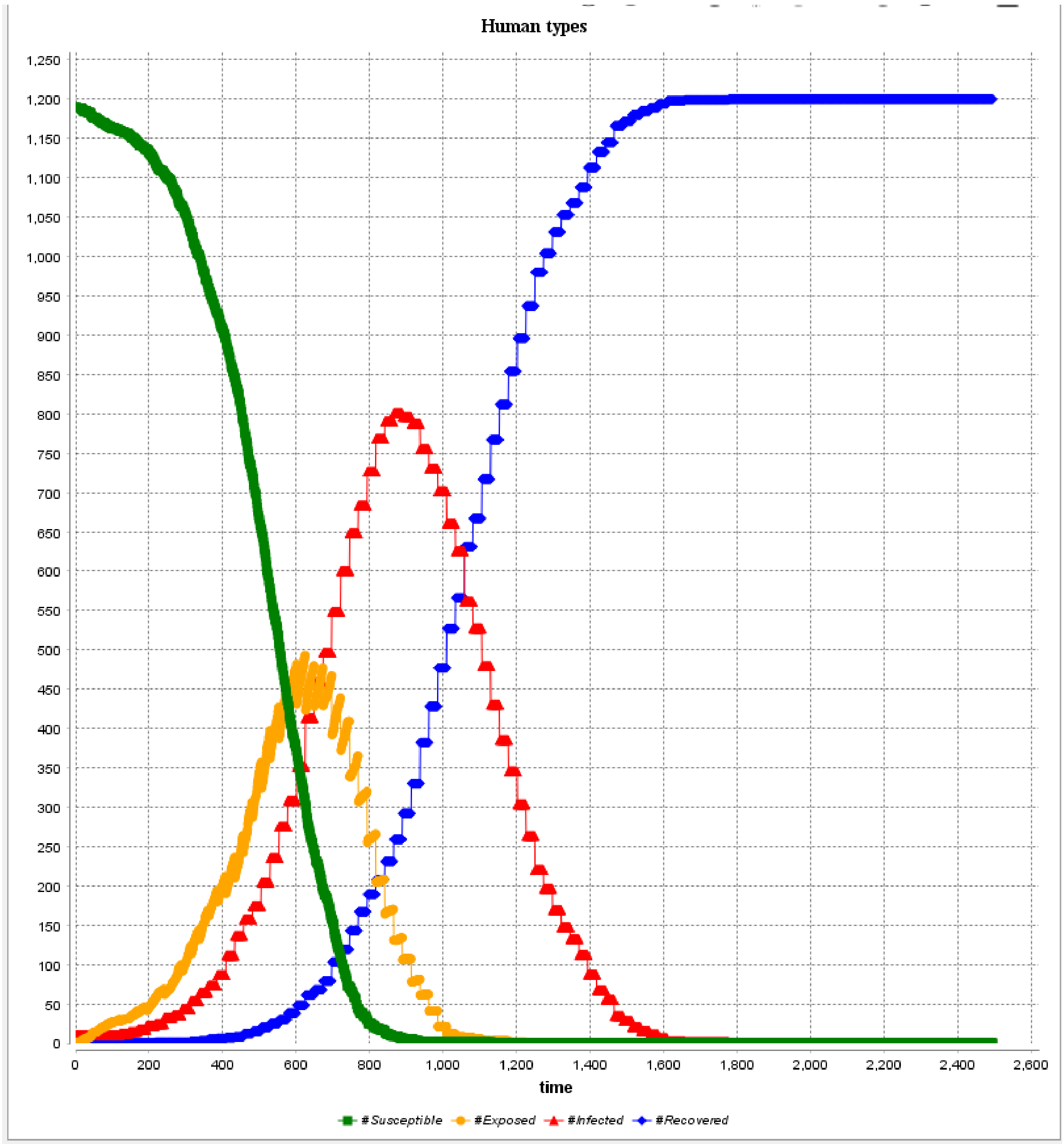
- For small population, the pandemic just dies out very quickly. Many people are not even exposed.

- For medium population, the pandemic infects more people, but some are unaffected.

- For large population, the pandemic infects everyone very quickly since the city is denser, leading to more contacts with infected.




Human types
400
390
380
370
360
350
340
330
320
310
300
290
280
270
260
250
240
230
220
210
200
190
180
170
160
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
0
0
200
400
600
800
1,000
1,200
1,400
1,600
1,800
2,000
2,200
2,400
2,600
time
#Susceptible
#Exposed
#Infected
#Recovered



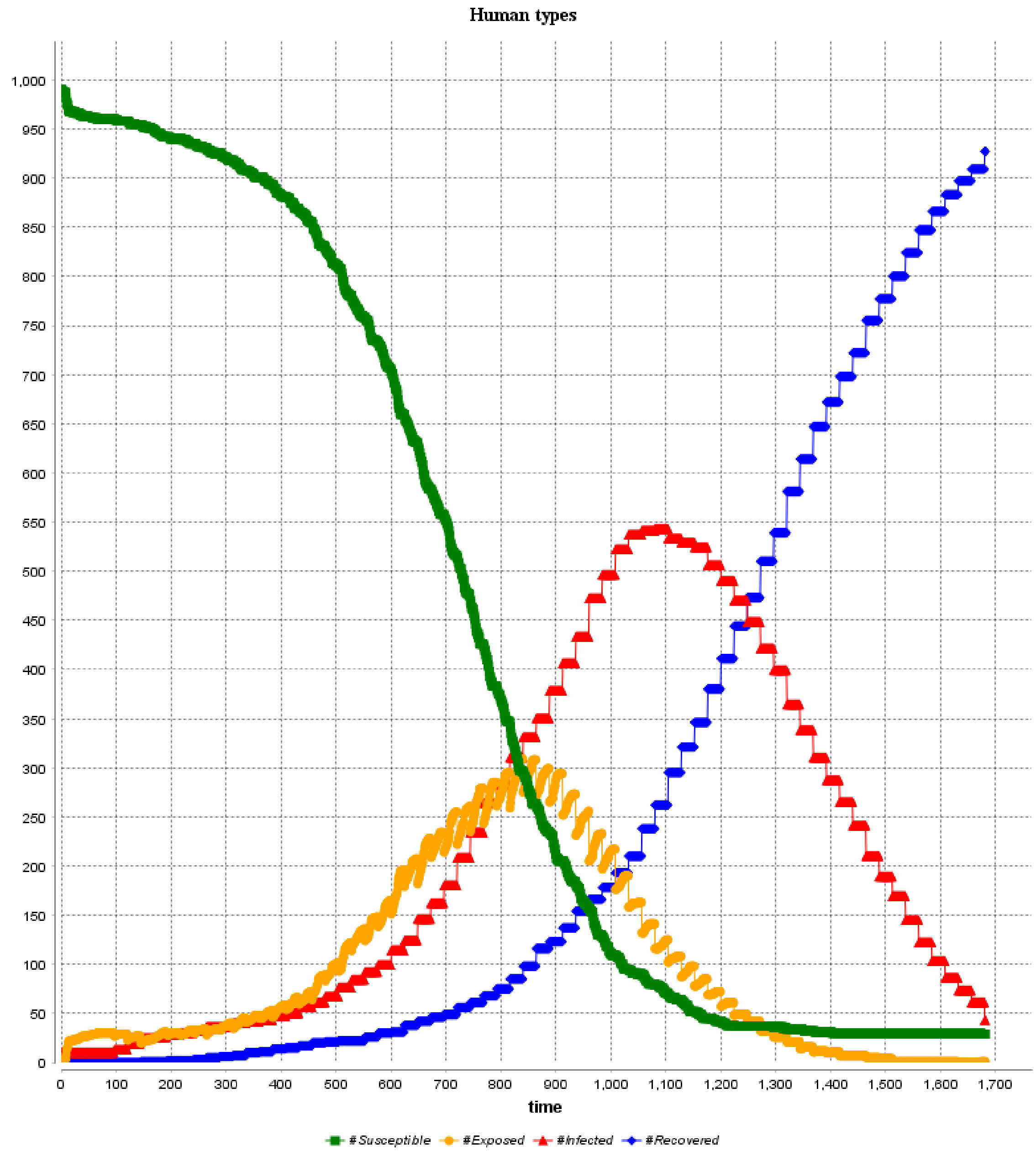
Human types
1,250
1,200
1,150
1,100
1,050
1,000
950
900
850
800
750
700
650
600
550
500
450
400
350
300
250
200
150
100
50
0
0
200
400
600
800
1,000
1,200
1,400
1,600
1,800
2,000
2,200
2,400
2,600
time
#Susceptible
#Exposed
#Infected
#Recovered

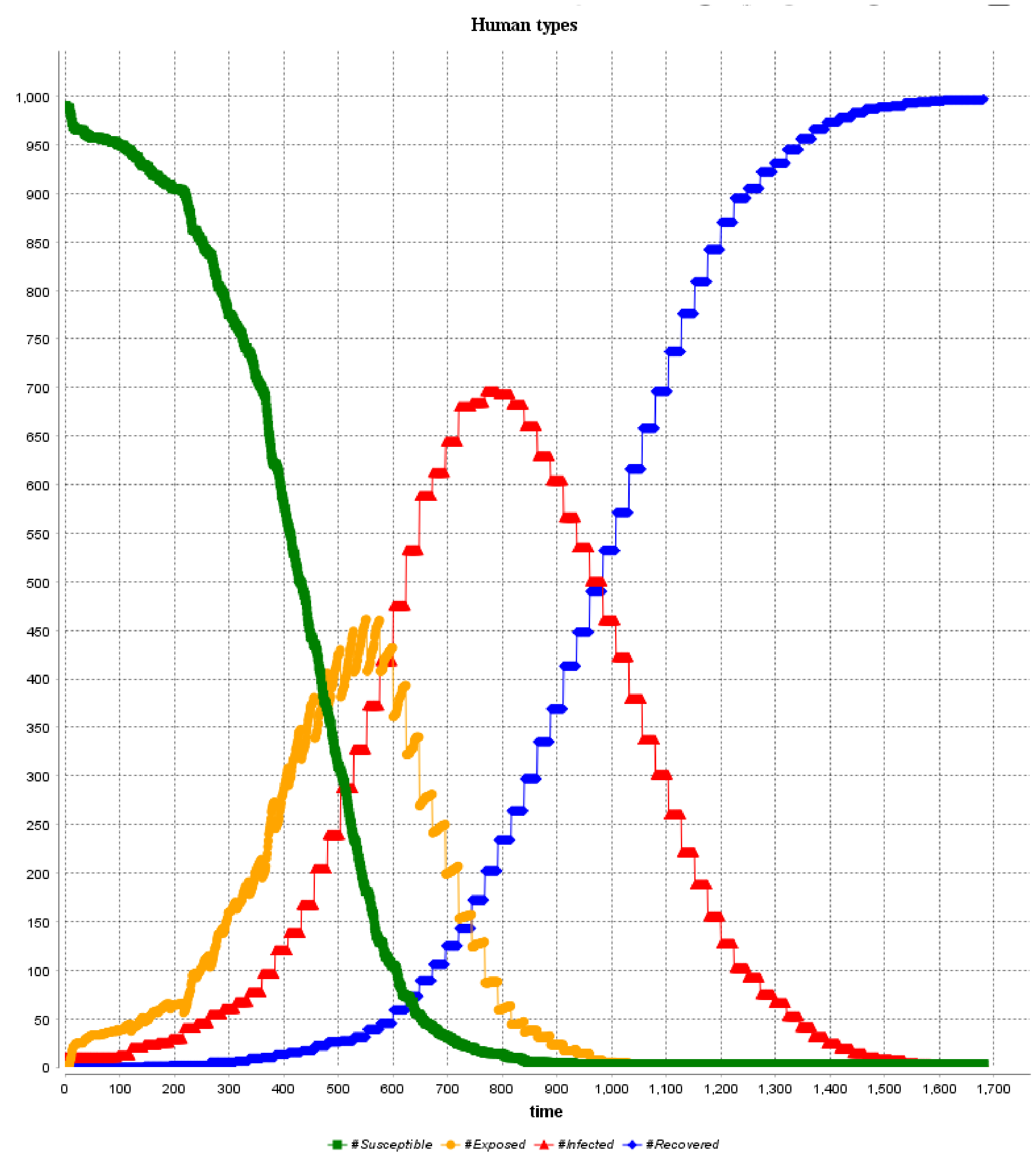
A.3 E1-3

- See file resultE1-3.csv

A.4 E2-4

- Without contagion in building, the pandemic takes longer to spread and has lower peak. With contagion in building, it spreads very quickly with a high peak.




Human types
1,000
950
900
850
800
750
700
650
600
550
500
450
400
350
300
250
200
150
100
50
0
0
100
200
300
400
500
600
700
800
900
1,000
1,100
1,200
1,300
1,400
1,500
1,600
1,700
time
#Susceptible
#Exposed
#Infected
#Recovered

A.5 Exploration E3-1

* Note: unless specified, people are not tested for the virus. We do this because random testing has a huge impact, making it harder to see the effects of other policies.

- In exploration E3, by default, the policies are applied after 5 days. In the experiment, you can change the date or apply policies after a certain number of infected by changing the parameters.

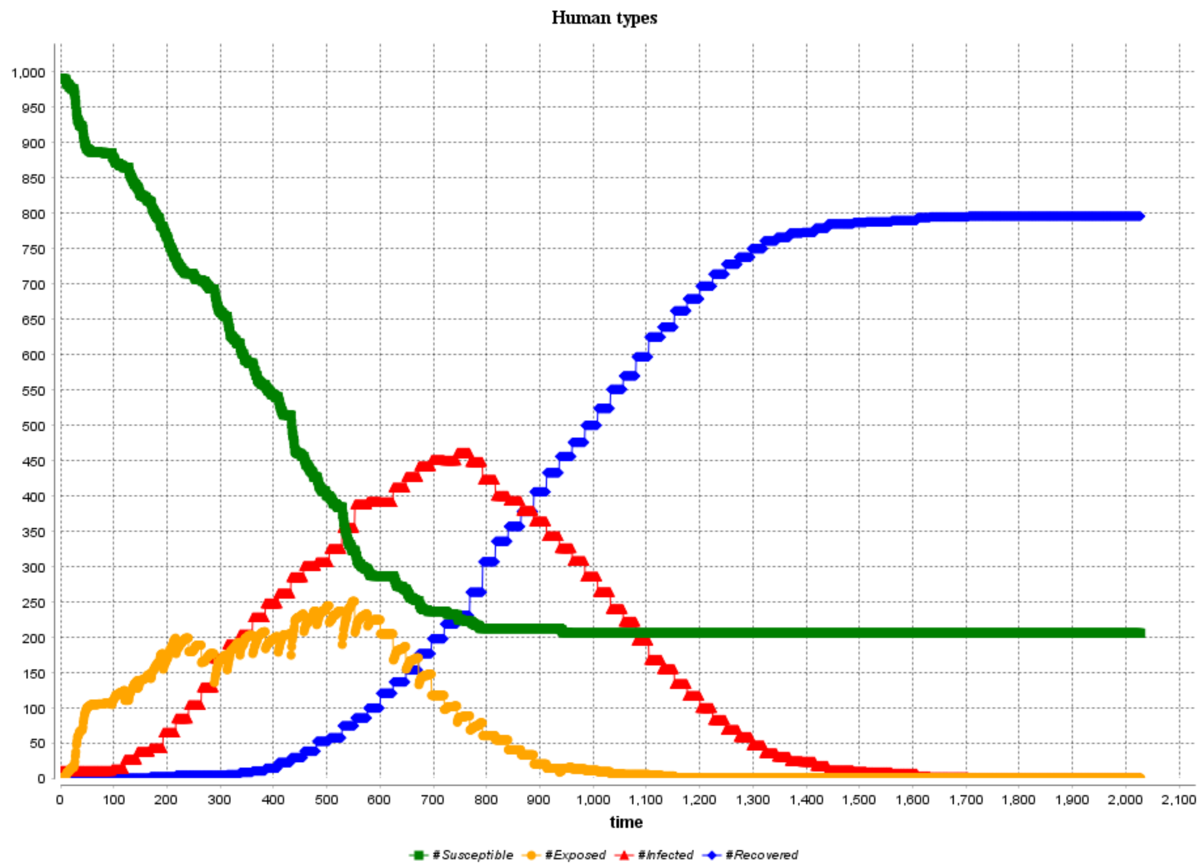


Figure 3. E3 no policy but has testing

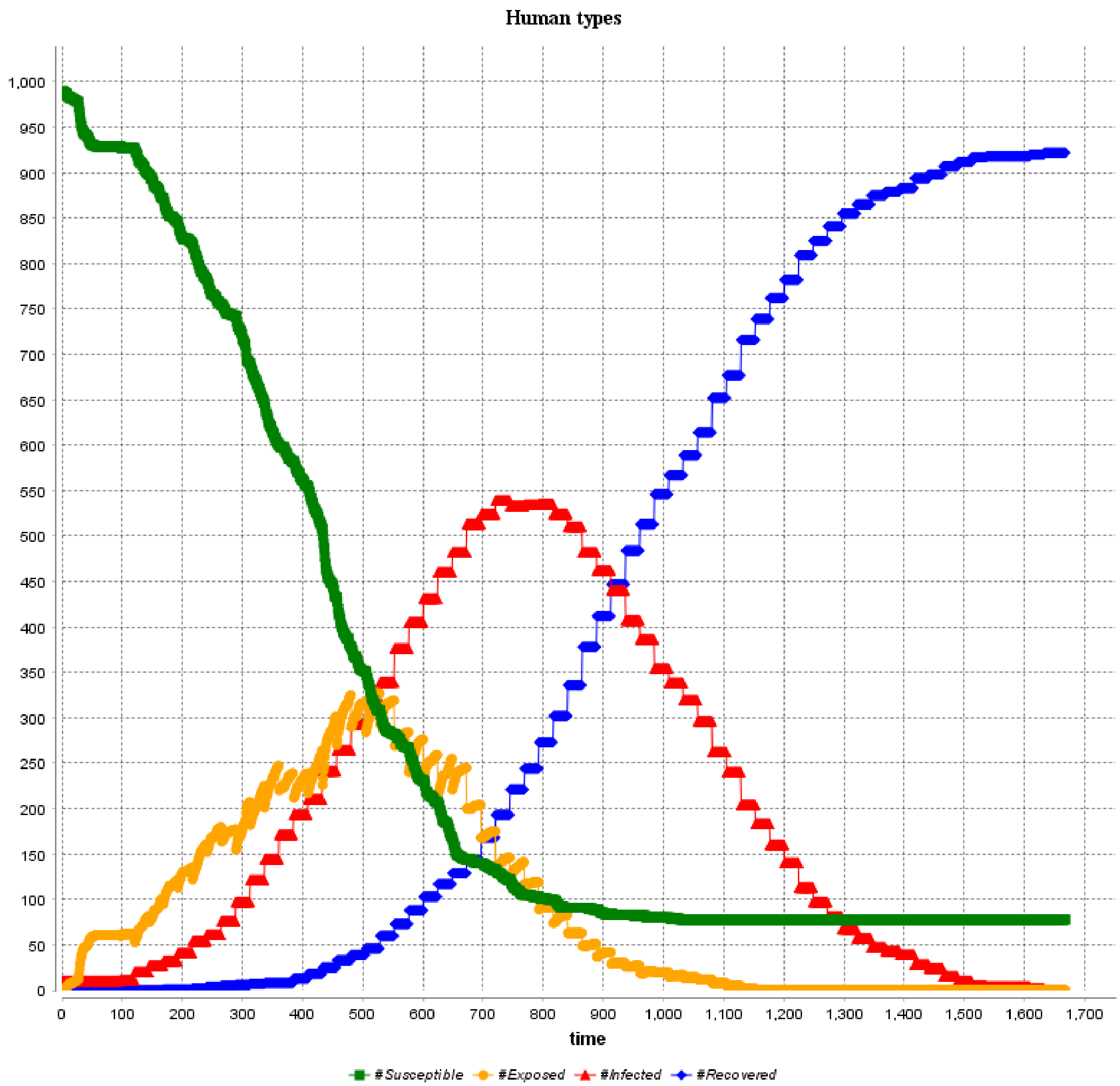


Figure 4. E3 no policy

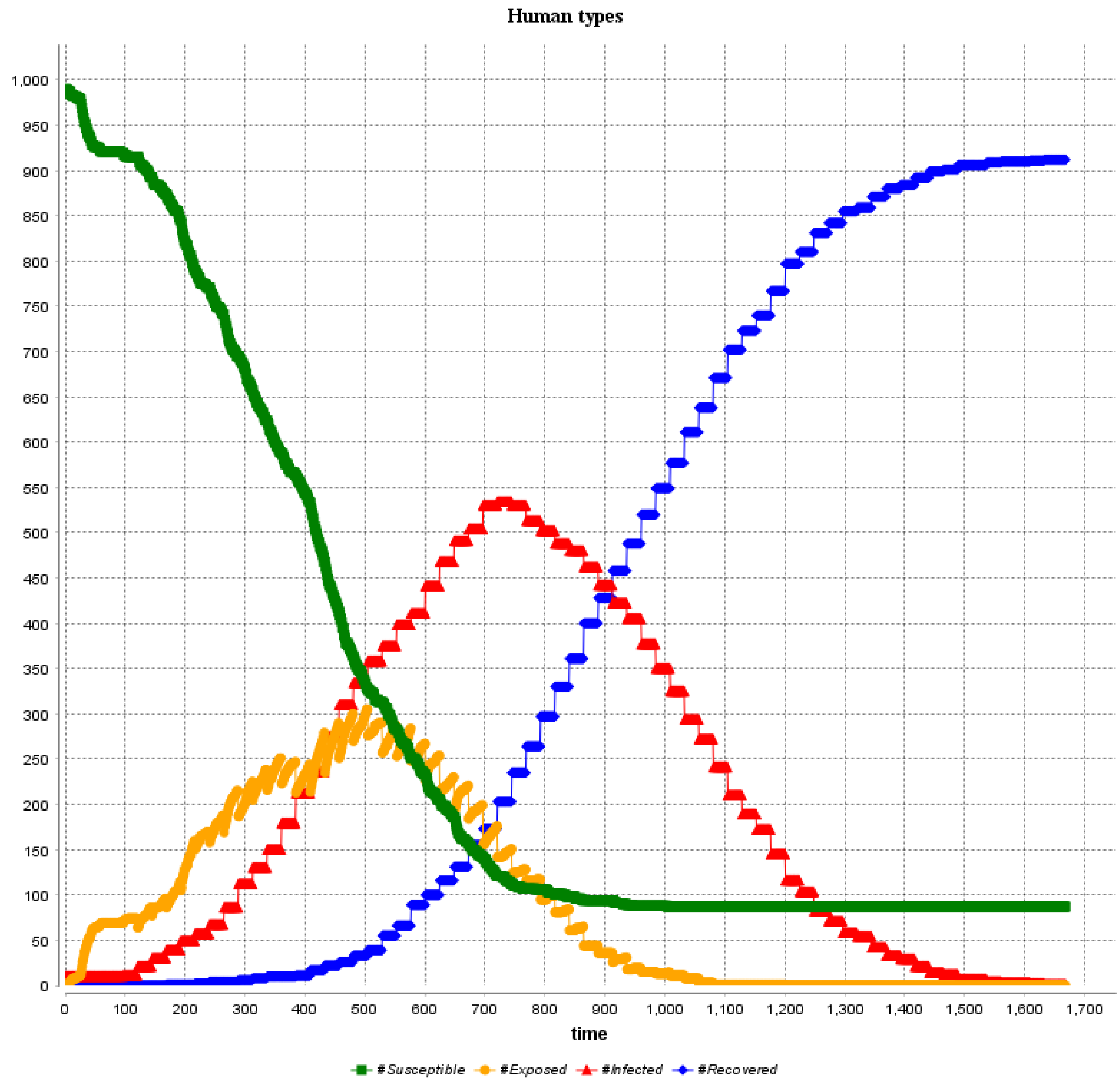


Figure 5. E3 wear mask

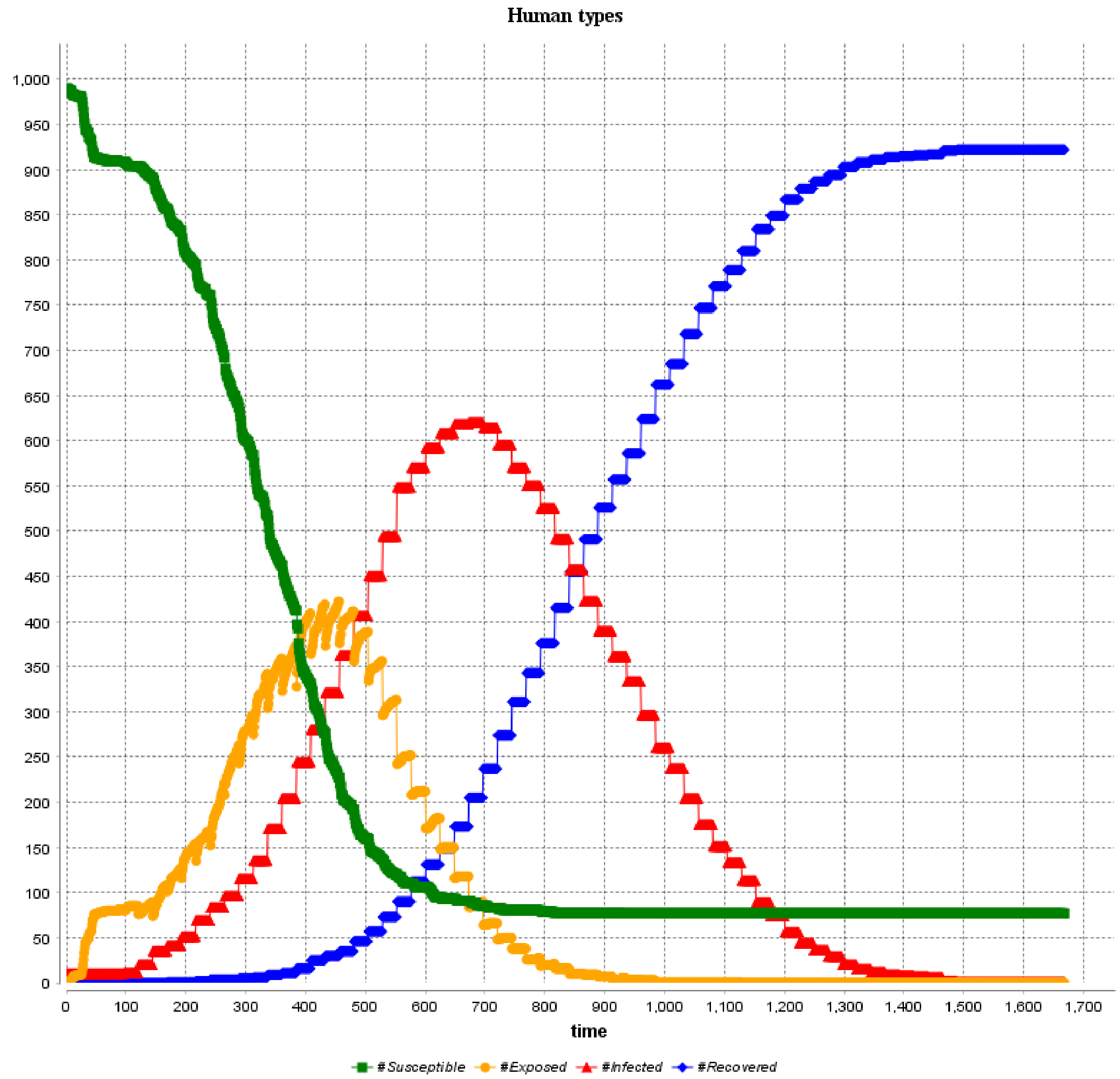


Figure 6. E3 wear mask, containment by age

- We see that wearing mask and containment by age has very little impact on the result of our experiment. This happens because our environmental transmission is a bit too aggressive.

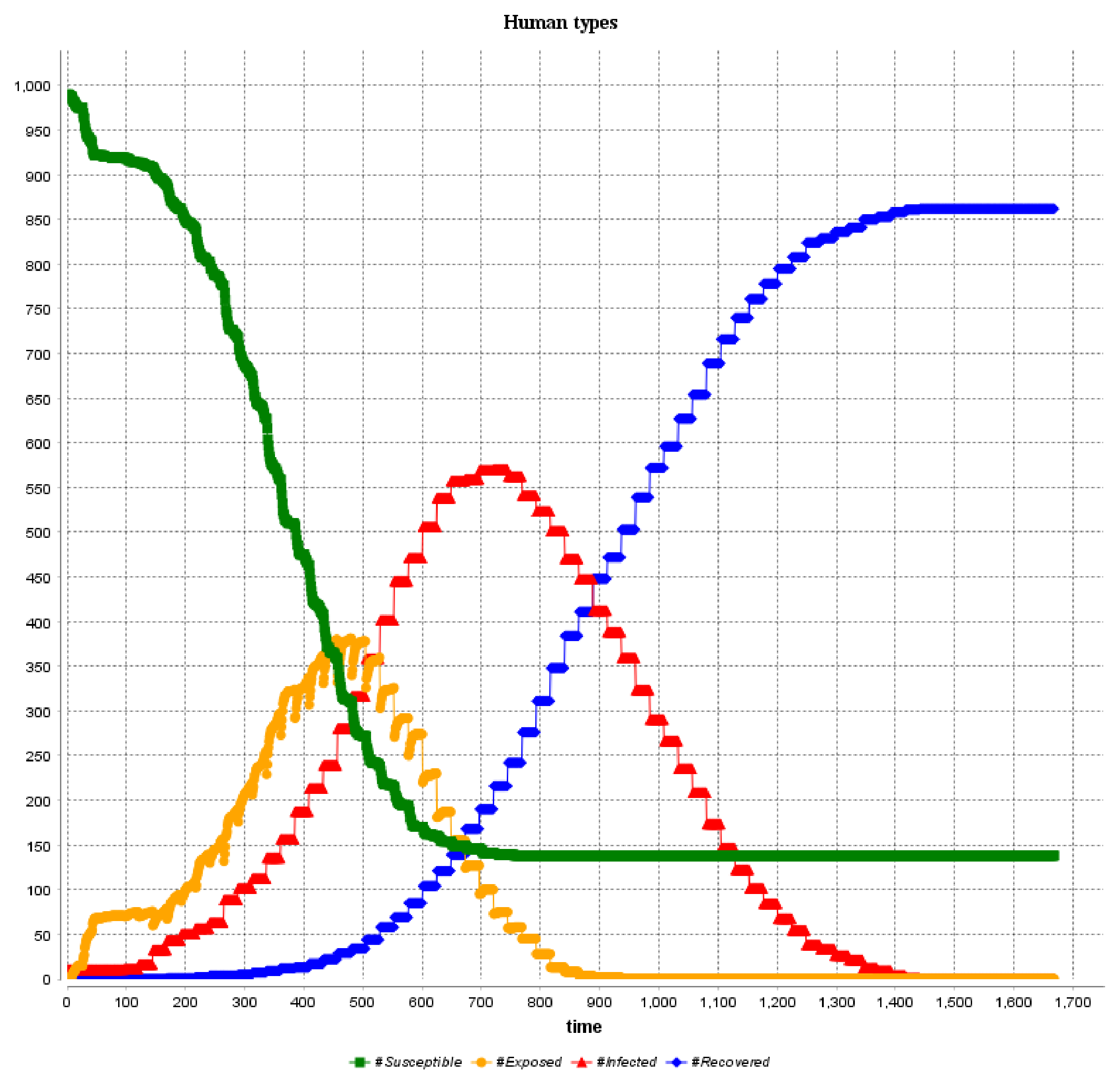


Figure 7. wear mask, containment by age, close schools

- Closing school has a small but noticeable impact on the result. However, since most of the population belong to the adult group, and the disease has high incubation time, the difference is still small.

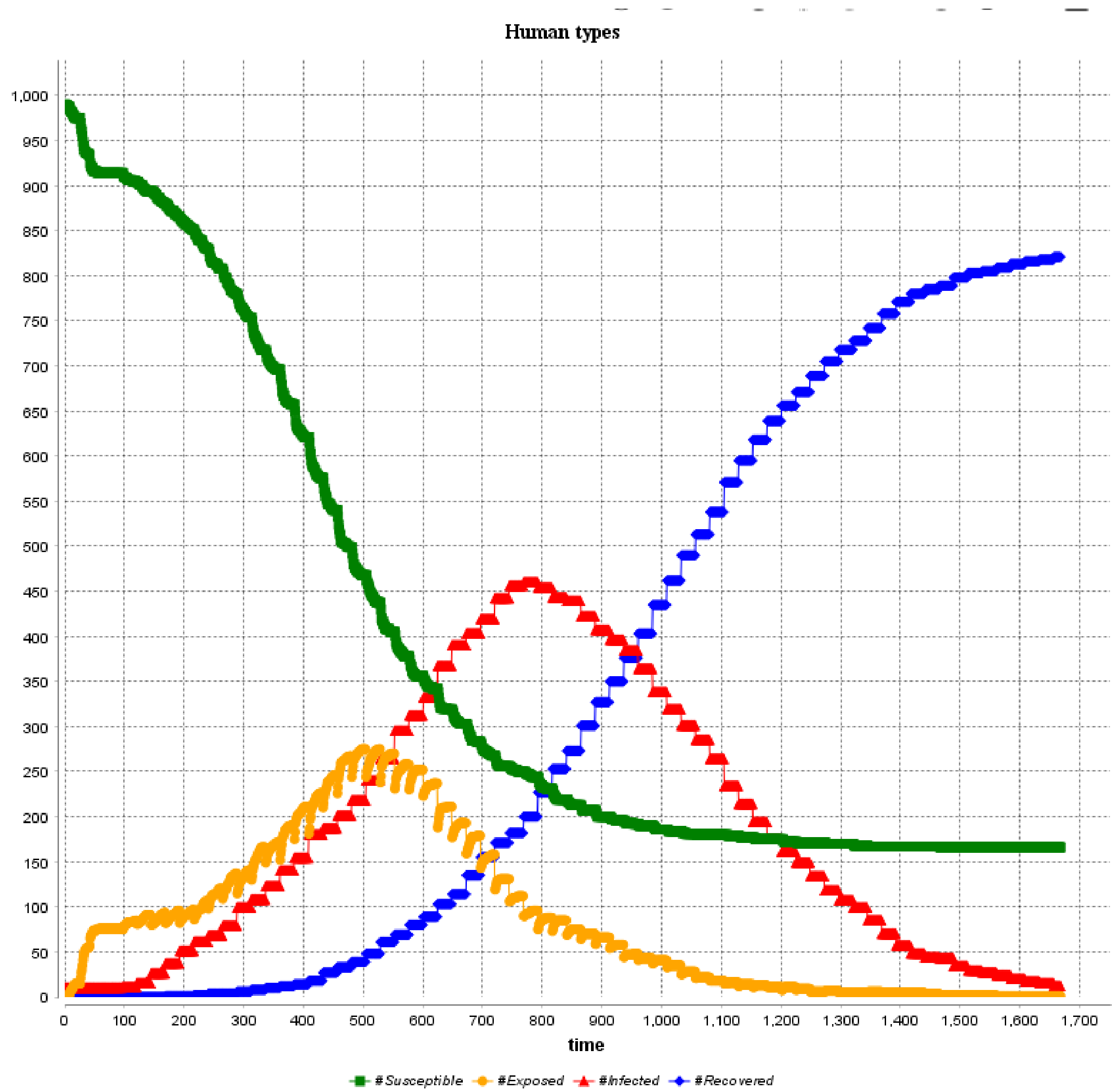


Figure 8. wear mask and total lockdown

- We can see that the curve is much flatter thanks to lockdown, and there are fewer infected people. However, since people still need to go buy food at supermarkets, over 75% are still infected.

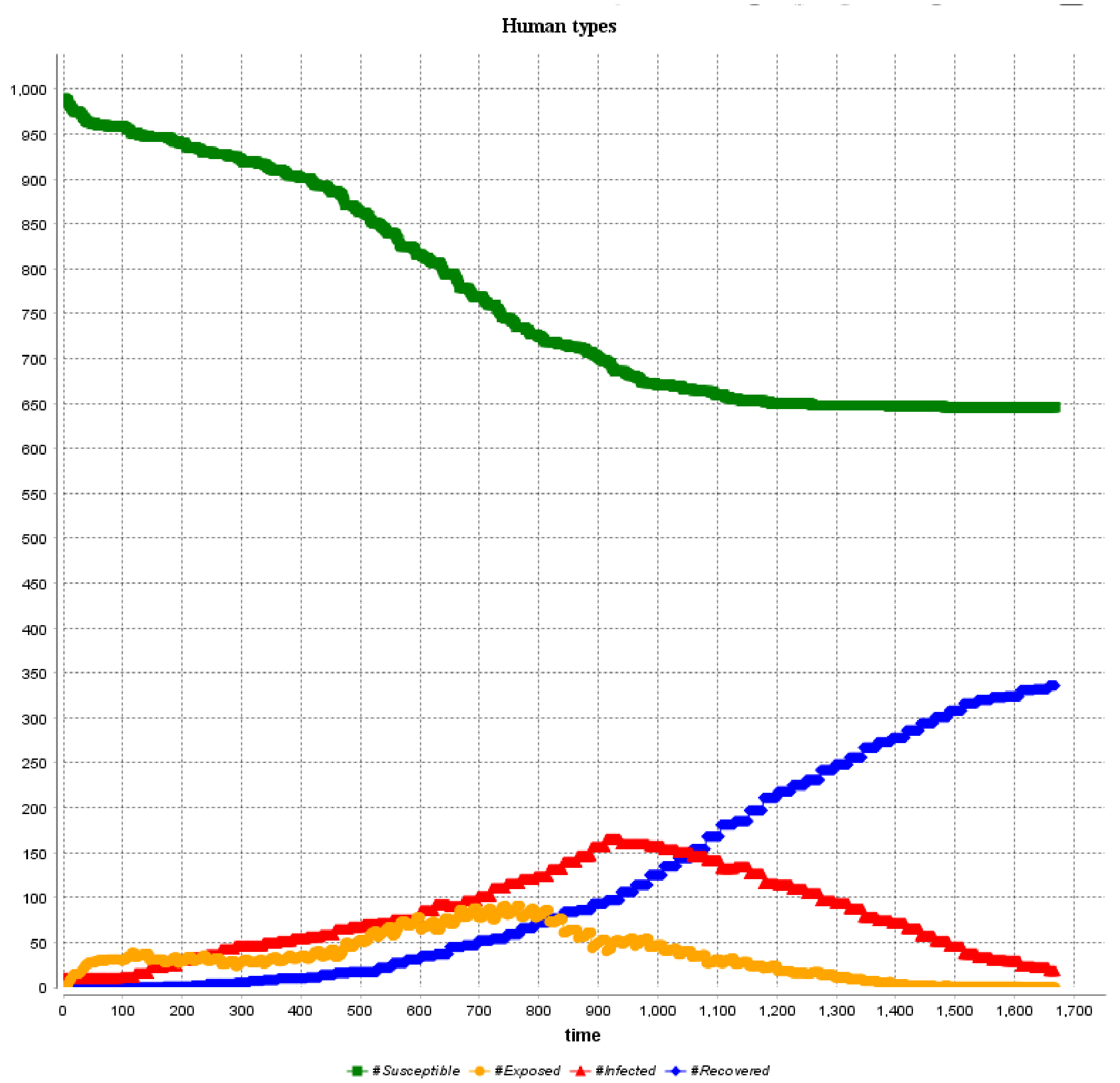


Figure 9. wear mask, total lockdown, free food delivery at home

- This graph shows the importance of “true lockdown”. Food is delivered so people do not have to go anywhere, reducing transmission chance. There’s still a small peak which is the result of early transmission.

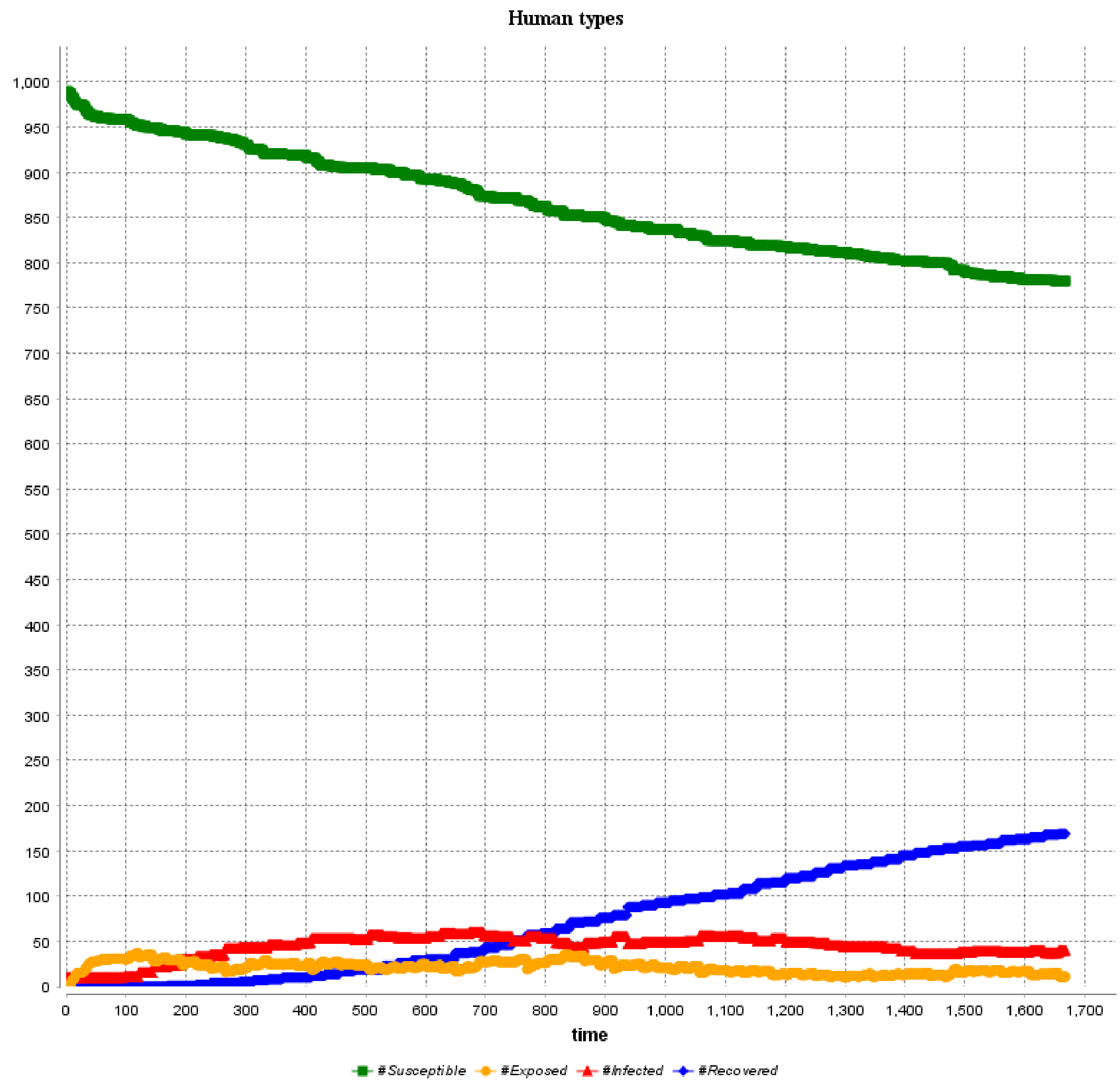


Figure 10. wear mask, total lockdown, free food delivery at home, random testing

- Finally, this is the result if early testing is done. When the pandemic is in the early stage, testing helps identify infected people and quarantine them immediately. There is no exponential growth of infection thanks to this policy. It is undoubtedly the policy with the greatest impact (when combined with lockdown).

B Annex

Consider the following parameter values for COVID19 epidemic dynamics.
These are simplifications and adaptation from literature.

- Probability of transmission (between human beings and from buildings to human beings): 17%
- Probability for an Exposed to become Asymptomatic: 30%
- Reduction of the probability transmission for an Asymptomatic: 45%
- Reduction of the transmission probability by masks: 50% - Incubation period (time in state E): from 3 to 10 days
- Infectious period (time in state I or Asymptomatic): from 10 to 30 days
- Tests: probability of true positive: 89% (i.e. a test as a probability of 89% to return that an individual is infected when it is infected).
- Tests: probability of true negative: 92% (i.e. a test as a probability of 92% to return that an individual is not infected when it is not infected).