

Report Modelling And Simulation Complex Model 2021

Subject: spread of COVID-19 in a city and
control policy

Author: Duc Dinh Anh
Email: ducda.m20ict@st.usth.edu.vn
Mobile: (+84) 982 086 651

July 2021

1 Spread of the disease

In this section, we want to discover the evolution of virus, beside we can track the steps of infectious process to see how and why virus can spread.

Model M1

Strategy

We build this model base on the SEIR strategy which following states:

- S: **Susceptible**, meaning the individual can be infected,
- E: **Exposed**, meaning that the individual has been infected, but cannot be infected other individuals,
- I: **Infectious**, meaning that the individual has been infected and can infect other individuals,
- R: **Recovered**, meaning that the individual has recovered from disease and cannot be infected anymore.

- ***Note:** S individual will be infected by I individual. E will become I from three to seven day and I will become R from seven to ten day.

Requirement

- Implement population of species "Individuals" with an attribute for each epidemic state.
- Each individual are moving randomly by using wander skill which is built in method of GAMA.
- At each state, I individual infects one of S individual.
- Create mechanism that allow E individuals automatically change their states to I.(Following the duration)
- Create mechanism that allow I individuals automatically change their states to R.(Following the duration)
- Displaying each individual with a circle and a colour depends on their state.
- Creates 500 individuals and one infected individual in population, plot number of species in each states.

Result

```
1 global{
2   int number_of_people <- 500;
3   int number_of_infected_people <- 1;
4   float dangerous_distance <- 0.5#m;
5   int pandemic_duration <- 0;
6   init{
7     create individuals number: number_of_people;
8     loop i from: 0 to: number_of_infected_people - 1{
```

```

9      ask one_of(individuals){
10         is_infected <- true;
11         my_color <- #yellow;
12         epidemic_state <- "E";
13         count_date_expose <- 1;
14     }
15 }
16 }
17 }

```

- In the global agent I create some attributes to calibrate the parameter of model and there are:

- number_of_people: allow me to add more individual.
- number_of_infected_people: allow me to add infected people.
- pandemic_duration: counting the date of pandemic duration.
- dangerous_distance: Is the perfect distance for virus go inside susceptible individual.

```

1 species individuals skills:[moving]{
2   bool is_infected <- false;
3   string epidemic_state <- "S";
4   rgb my_color <- #blue;
5   int count_date_expose <- 0;
6   int count_date_infectious <- 0;
7
8   reflex move{
9     do wander speed: 1.0;
10  }
11
12  //Expose to infectious
13  reflex dynamicTurnBad when:(count_date_expose >= 72) and (count_date_expose <=
14    write "Change bad state";
15    epidemic_state <- "I";
16    is_infected <- true;
17    my_color <- #red;
18    count_date_infectious <- count_date_expose + 1;
19  }
20
21  //Infectious to recovery
22  reflex dynamicTurnGood when: (count_date_infectious >= 240) and (count_date_in
23    write "Change good state";
24    epidemic_state <- "R";
25    is_infected <- false;
26    my_color <- #green;
27    count_date_expose <- 0;
28    count_date_infectious <- 0;
29  }

```

```

30
31   reflex infect when:(epidemic_state = "I"){
32     ask individuals at_distance 3.0 {
33       if (self.epidemic_state = "S"){
34         self.is_infected <- true;
35         self.epidemic_state <- "E";
36         self.my_color <- #yellow;
37       }
38     }
39   }
40   //Counter of expose duration
41   // one day is 24 hour -> 3 days is 72 hour
42   // one day is 24 hour -> 10 days is 240 hour
43   // one day is 24 hour -> 30 days is 720 hour
44   reflex increaseExposeDate when: epidemic_state = "E"{
45     if(count_date_expose = 240){
46       count_date_expose <- 0;
47     }
48     if(cycle mod 60 = 0 and cycle != 0){
49       count_date_expose <- count_date_expose + 1;
50     }
51   }
52
53   reflex increaseInfectiousDate when: epidemic_state = "I"{
54     if(count_date_infectious = 720){
55       count_date_infectious <- 0;
56     }
57     if(cycle mod 60 = 0 and cycle != 0){
58       count_date_infectious <- count_date_infectious + 1;
59     }
60     pandemic_duration <- count_date_infectious;
61   }
62
63   aspect infor{
64     draw circle(0.5) color: my_color;
65   }
66 }

```

- In the individual agent I create some attributes to contain the value that can control the model follows my scenario:
- is_infected: separating people who is infected or not infected.
- epidemic_state: point out the state of each individual.
- my_color: color of each individual which can be decided by it's state.
- count_date_expose: counting the E duration of each individual.
- count_date_infectious: counting the I duration of each individual.

- This agent have five actions an each action have their own function base on the requirement.
- "reflex infect": This action allows individual infects for susceptible individual. In this action, I will pick one Infectious individual and use this one to find the other individuals which has the same distance with the value of dangerous_distance variable.
- "reflex dynamicTurnBad" and "reflex increaseExposeDate": These two actions connect to each other, the second action counts the date when an individual stays in E state, and it will automatically change to I state when the condition of the first action satisfies, that means the state of individual will remain at E until it meets the end of the duration after that it will become I state.
- "reflex dynamicTurnGood" and "reflex increaseInfectiousDate": These two actions connect to each other, the second action counts the date when an individual stays in I state, and it will automatically change to R state when the condition of the first action satisfies, that means the state of an individual will remain at I until it meets the end of the duration after that it will become R state.

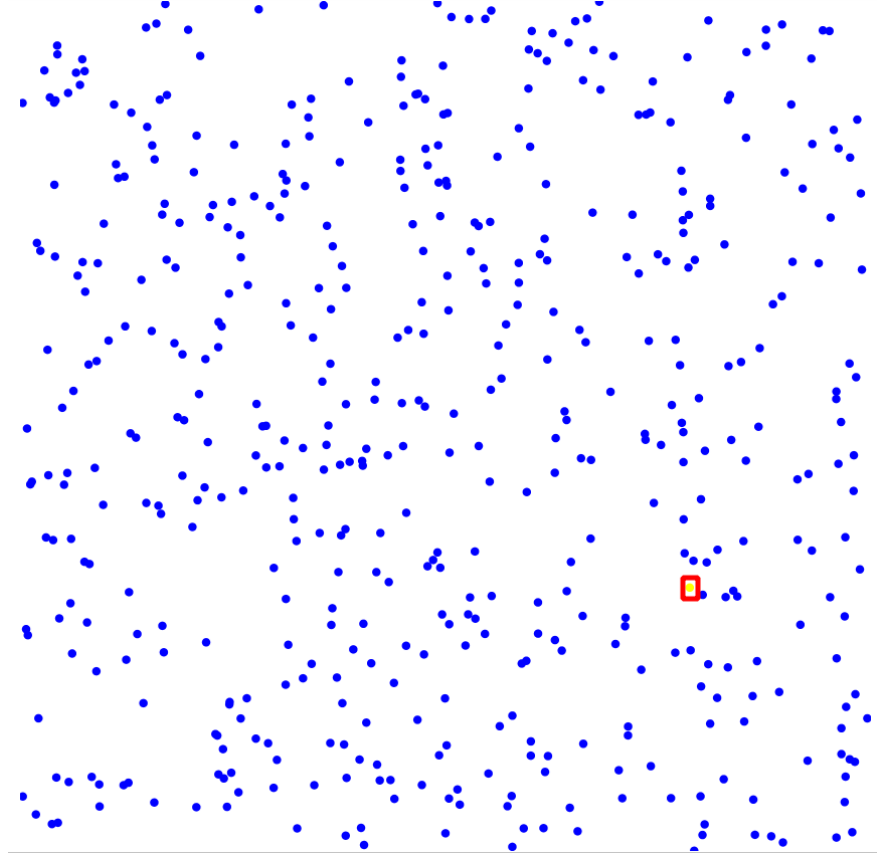


Figure 1: Simulation of model M1_1, following the requirement 1 infected individual and 500 susceptible individuals.

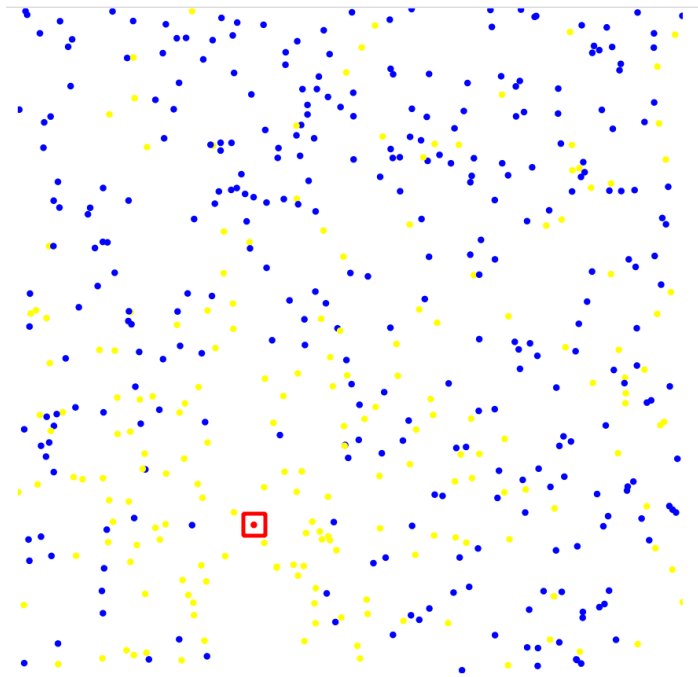


Figure 2: Simulation of model M1.1, after three days E individual turn to state I and start spreading disease to other individuals.

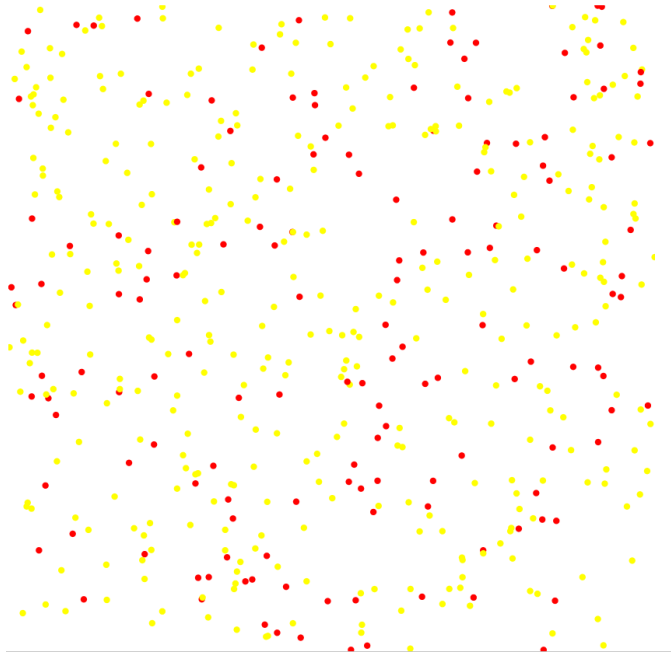


Figure 3: Simulation of model M1.1, all E individuals change to I state.

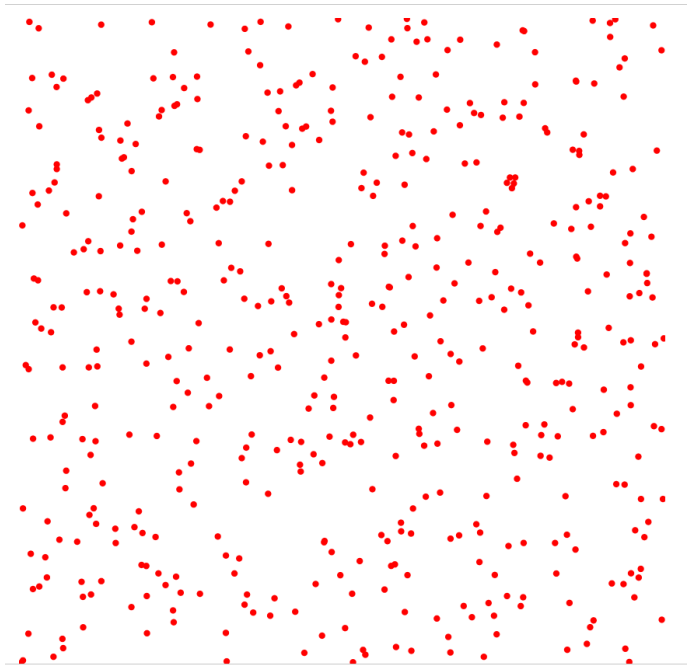


Figure 4: Simulation of model M1.1, all E individuals become infected individual.

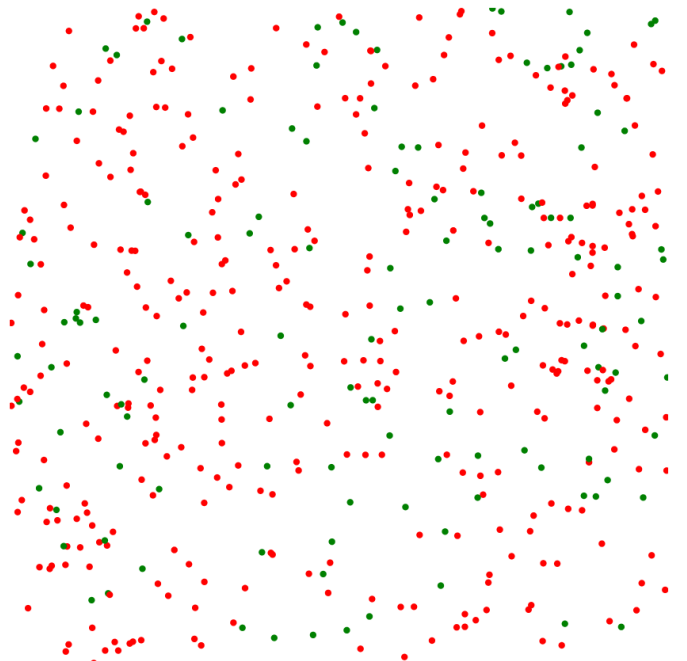


Figure 5: Simulation of model M1.1, all I individuals gradually change to recovery individual.

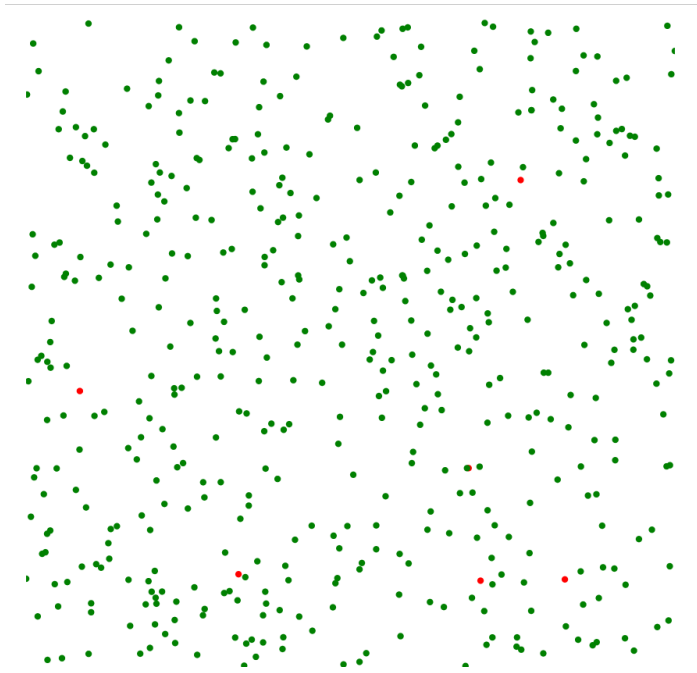


Figure 6: Simulation of model M1.1, number of individuals who have I state increase very fast.

Exploration

Exploration E1.1. Effect of randomness:

Define a gui experiment, creating 10 experiments (with different seed values) and plot the evolution of the number of I individuals for each simulation.

```
1 experiment E1_1 type: gui{
2   parameter "Number of people: " var: number_of_people min:10
3   max: 1000 step: 10;
4   parameter "Number of infected people: "
5   var: number_of_infected_people min: 1 max: 50 step: 10;
6   init{
7     create simulation with:[number_of_people::number_of_people ,
8     number_of_infected_people::number_of_infected_people ,seed::1];
9     create simulation with:[number_of_people::number_of_people ,
10    number_of_infected_people::number_of_infected_people ,seed::2];
11    create simulation with:[number_of_people::number_of_people ,
12    number_of_infected_people::number_of_infected_people ,seed::3];
13    create simulation with:[number_of_people::number_of_people ,
14    number_of_infected_people::number_of_infected_people ,seed::4];
15    create simulation with:[number_of_people::number_of_people ,
16    number_of_infected_people::number_of_infected_people ,seed::5];
17    create simulation with:[number_of_people::number_of_people ,
18    number_of_infected_people::number_of_infected_people ,seed::6];
19    create simulation with:[number_of_people::number_of_people ,
20    number_of_infected_people::number_of_infected_people ,seed::7];
21    create simulation with:[number_of_people::number_of_people ,
22    number_of_infected_people::number_of_infected_people ,seed::8];
23    create simulation with:[number_of_people::number_of_people ,
24    number_of_infected_people::number_of_infected_people ,seed::9];
25    create simulation with:[number_of_people::number_of_people ,
26    number_of_infected_people::number_of_infected_people ,seed::10];
27  }
28  output{
29    display E1_1{
30      chart "Ch" type:series{
31        datalist ["#S", "#E", "#I", "#R"]
32        value: [individuals count (each.epidemic_state = "S"),
33        individuals count (each.epidemic_state = "I"),
34        individuals count (each.epidemic_state = "E"),
35        individuals count (each.epidemic_state = "R")]
36        color: [#blue, #yellow, #red, #green];
37      }
38    }
39  }
40 }
```

- I design an experiment that can simulate ten simulations one time, and each simulation have different seed like the requirement. In this experiment I want to discover the evolution of each state base on the number of individual.

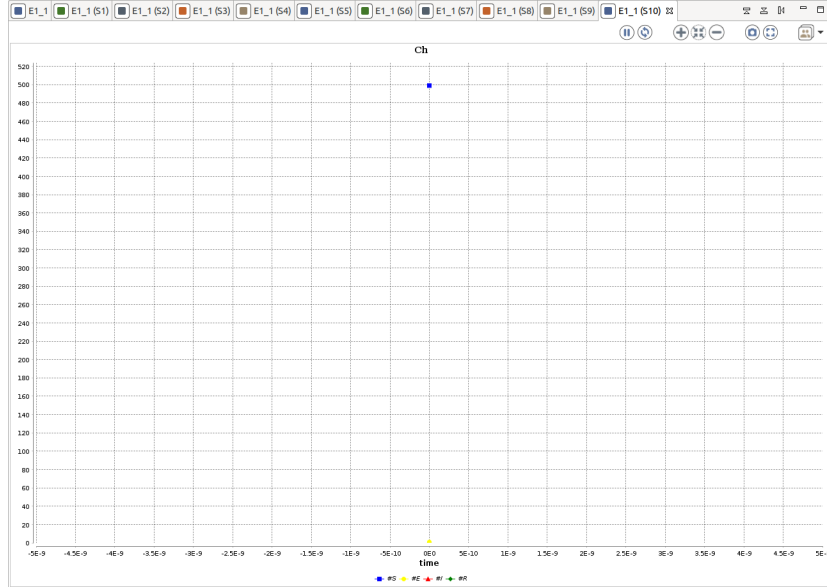


Figure 7: Exploration E1_1, ten experiments.

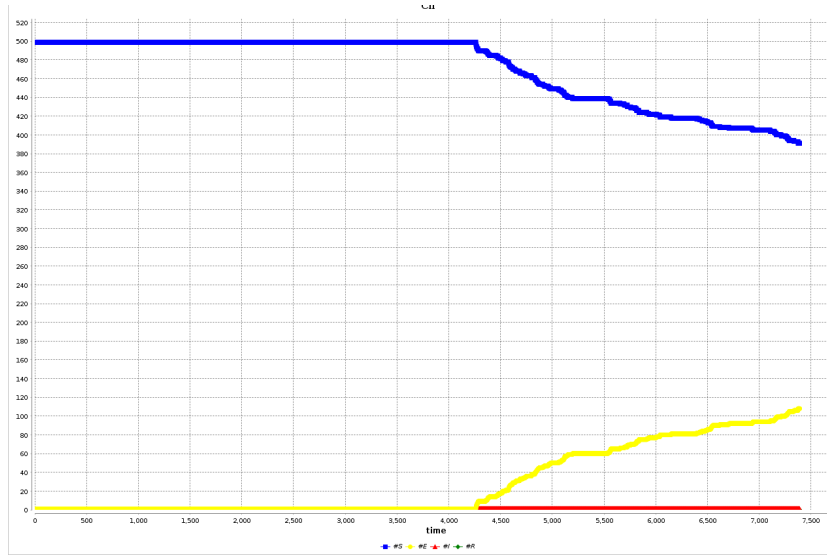


Figure 8: Exploration E1_1, E cases increase, S cases decrease.

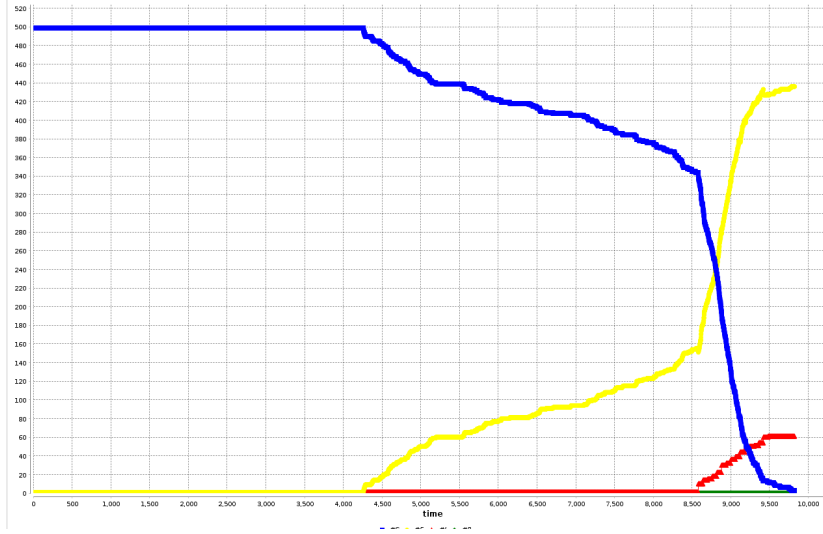


Figure 9: Exploration E1.1, I cases appear.

Exploration E1.2. Impact of the number of individuals. Define a gui experiment, creating 11 experiments, with the same seed value but with a number of individuals from 200 to 2000 (with a step of 200) and plot the evolution of the number of I individuals.

```

1 experiment E1.2 type: gui{
2   parameter "Number of people: "
3   var: number_of_people min:200 max: 2000 step: 200;
4   parameter "Number of infected people: "
5   var: number_of_infected_people min: 1 max: 50 step: 10;
6
7   init{
8     create simulation with:[number_of_people::number_of_people ,
9     number_of_infected_people::number_of_infected_people , seed:: seed ];
10    create simulation with:[number_of_people::number_of_people ,
11    number_of_infected_people::number_of_infected_people , seed:: seed ];
12    create simulation with:[number_of_people::number_of_people ,
13    number_of_infected_people::number_of_infected_people , seed:: seed ];
14    create simulation with:[number_of_people::number_of_people ,
15    number_of_infected_people::number_of_infected_people , seed:: seed ];
16    create simulation with:[number_of_people::number_of_people ,
17    number_of_infected_people::number_of_infected_people , seed:: seed ];
18    create simulation with:[number_of_people::number_of_people ,
19    number_of_infected_people::number_of_infected_people , seed:: seed ];
20    create simulation with:[number_of_people::number_of_people ,
21    number_of_infected_people::number_of_infected_people , seed:: seed ];
22    create simulation with:[number_of_people::number_of_people ,
23    number_of_infected_people::number_of_infected_people , seed:: seed ];

```

```

24     create_simulation with:[number_of_people::number_of_people ,
25     number_of_infected_people::number_of_infected_people ,seed::seed];
26     create_simulation with:[number_of_people::number_of_people ,
27     number_of_infected_people::number_of_infected_people ,seed::seed];
28   }
29   output{
30     display E1_2{
31       chart "Ch" type:series{
32         datalist ["#S", "#E", "#I", "#R"]
33         value: [individuals count (each.epidemic_state = "S"),      individu
individuals count (each.epidemic_state = "I"),      individuals count (each.ep
color: [#blue, #yellow, #red, #green];
34     }
35   }
36
37 }
38 }

```

2 Spread in a city with a heterogeneous population

3 Public health policy