

Notes on Simulating COVID Testing*

Glenn Ellison[†]

June 2020

Abstract

1 Introduction

With the aim of discussing the impact of allowing students to interact more freely with small groups of friends I'll discuss here how one can simulate the effectiveness of testing procedures with a realistic model of disease progression.

I'll start here with the simpler model where students are not arranged in groups and interact with others in an infinite population in a completely independent uniform model.

We will simulate the evolution of a disease outbreak that starts with a single infected individual, keeping track of an ever increasing population of affected individuals that that has N_t members. We will simulate in continuous time. In each simulation we will either find that the infection has stopped expanding at some time T , or we will see that N_t seems to be growing without bound at which point we'll stop the simulation and conclude that the zero infection state is unstable given the parameters.

2 Describing the system state

At each point in time, we will keep track of a matrix giving ten things about each individual, two of which describe actual current characteristics, and the rest of which describe how the individual's situation will change in the future. Using i as the index for individual i the current attribute/time matrix A^t at t is an $N_t \times 9$ matrix with the element in each column given by:

*

[†]Department of Economics, Massachusetts Institute of Technology, Cambridge MA 02139 and NBER, e-mail: gellison@mit.edu

1. $a_{i1}^t \in \{S, E, I, I^s, R\}$ describes the individual's disease state with the letters representing Susceptible, Exposed, Infectious, Infectious and symptomatic, and Recovered.
2. $a_{i2}^t \in \{\text{Free}, \text{SelfQuarantine}, \text{Quarantine}\}$ describing whether the individual is freely interacting, in a (presumably less strict) self-quarantine, or in an enforced quarantine following a positive test.
3. $a_{i3}^t \in \{\text{Unidentified}, \text{Identified}\}$ indicating whether i will be identified by contact tracers when the contact tracing of the next person who came in contact with i and has already tested positive is completed. This will occur at time a_{i10}^t .
4. $a_{i4}^t \in \{\text{Known}, \text{Waiting}^-, \text{Waiting}^+\}$ giving the results of the most recent test conducted on individual i . This indicates whether the test result is already known, whether it has not yet come back and will be negative when it does, or has not yet come back and will be positive when it does.
5. $a_{i5}^t \in \{\text{NA}, I^s, R\}$ indicating whether the next state that a currently infected person will transition to is I^s or R . The NA option is used for all individuals who are not currently infectious.
6. $a_{i6}^t \in \mathbb{R} \cup \infty$ is a real number giving the time at which the individual i 's disease will progress to the next step along the S, E, I, I^s, R path. It is set to ∞ when the individual is already recovered.
7. $a_{i7}^t \in \mathbb{R} \cup \infty$ is a real number giving the time at which individual i will next spread the disease to a new person. This is ∞ if individual i is not currently infectious, or if they will be quarantined or recover at the point at which the infection would have occurred.
8. $a_{i8}^t \in \mathbb{R} \cup \infty$ is a real number giving the time at which individual i will next be tested under the normal testing schedule. It goes to ∞ when individual i has already been permanently quarantined or has recovered.
9. $a_{i9}^t \in \mathbb{R} \cup \infty$ is a real number giving the time at which the most recent test result on individual i will be returned.
10. $a_{i10}^t \in \mathbb{R} \cup \infty$ is a real number giving the time at which individual i will next be identified as the contact of someone who has tested positive.

We also need to keep a second data structure giving all of the details of the tree that describes who infected whom. Individual 1 is at the top of the tree. Individual 2 is a branch below individual 1 because 2 got infected by 1. Individual 3 will either be another branch coming off from 1 or a branch coming off from 2 depending on whether 3 was infected by 1 or 2, and so on.

3 Parameters

The simulations will be carried out fixing a large number of parameters. Some of these parameters are:

- $\gamma_{EI}, \gamma_{II_s}, \gamma_{IR}, \gamma_{IsR}$ - describe the speed at which transitions from one disease state to the next occur.
- β - the rate at which infectious people infect others.
- p_{test} - the probability that someone who is positive will test positive.
- d_{test} - the length of time for test results to come back.
- Δ - the interval between times when each individual is periodically tested.
- $p_{contact}$ - the probability that contact tracing of an infected person will identify people who were infected by them.
- $\mu_{contact}, \sigma_{contact}$ - mean and variance of the the length of time for tracing of a contact to occur.
- ρ_{SQ} - factor by which infection rates are multiplied when someone is in self quarantine.

4 Algorithm overview

This matrix A^t starts as a 1×10 vector. It is initialized by setting $a_{1,1:5} = (E, \text{Free}, \text{Unidentified}, \text{Known}, \text{NA})$. Draw a_{16} from an exponential distribution with mean $1/\gamma_{EI}$. Draw a_{17} from an exponential distribution with mean $1/\beta$. Draw a_{18} from a uniform distribution on $[0, \Delta]$. Set $a_{19} = \infty$. Set $a_{110} = \infty$.

Over time A^t grows to a $10 \times N_t$ matrix. The basic algorithm by which it is updated can be described as:

1. Check if the epidemic has ended. This occurs if all individuals are either recovered or in permanent quarantine, i.e. they have $a_{i1} = R$ or $a_{i2} = \text{Quarantine}$. (In future analyses where susceptible people are put in the matrix having $a_{i1} = S$ will also be OK.) If so, end the analysis and record the number of people who were ever infected.
2. Check if the epidemic has reached the upper bound for the largest size we will track (perhaps 1000 or 10000). If the epidemic reaches this size, again end the analysis and record the number of people who were infected.
3. Otherwise, look for the next event that will occur. This is identified by finding the smallest number in the submatrix consisting of rows 6 through 10. Advance the clock to the time of this event. Process the event as appropriate updating the matrix. Then go back to step 1.

Each simulation ends at a time T and we record the number of people N_T that were ever infected in the epidemic. By running a number of simulations for a given set of parameters we'll estimate the distribution of the number of people who will be infected. We'll be particularly interested how this changes with Δ and what value of Δ is needed to keep epidemics from exploding.

5 Algorithm details

In each round of the algorithms main loop we identify a single individual i who has the earliest event. We update the clock to this time, and process the event, which may change entries in column i , entries in other columns, and add a column to the table. The details depend on what type of event occurs. I'll go through them one type of event at a time.

- Disease progression (a_{i6}).

If $a_{i1} = E$, then advance to I . If $a_{i1} = I$, then advance to I^s or R depending on the value of a_{i5} . If $a_{i1} = I^s$, then advance to R .

We also update the parts of i 's state describing the next disease evolution. Let the current time be t_0 . If the new state is I^s , then we draw x from an exponential distribution with mean $1/\gamma_{IsR}$ and set $a_{i6} = t_0 + x$. If the new state is I , then we draw two new random variables, $y \sim \text{Exp}(1/\gamma_{IIs})$ and $z \sim \text{Exp}(1/\gamma_{IR})$, set a_{i6} equal to the smaller of $t_0 + y$ and $t_0 + z$, and set a_{i5} equal to Is or R to indicate which transition occurred earlier.

If the new value of a_{i1} is I , then we also update the time at which i will cause a new infection. Draw w from a distribution that exponential with mean $1/\beta$ if $a_{i2} = \text{Free}$ or mean rhosQ/beta if $a_{i2} = \text{SelfQuarantine}$, and set $a_{i7} = t_0 + w$. Set $a_{i7} = \infty$ if $a_{i2} = \text{Quarantine}$.

If the new value of $a_{i1} = R$, then we update the time at which i will cause a new infection to ∞ .

- New infection (a_{i7})

If the next event is that i will cause a new infection at t_0 , then we increase N_t by one and add a new row to the matrix. We initialize the row of the matrix similarly to how we initialized the first row. We set $a_{N_t,1:5} = (E, \text{Free}, \text{Unidentified}, \text{Known}, \text{NA})$. Draw $a_{N_t,6}$ from an exponential distribution with mean $1/\gamma_{EI}$. Draw $a_{N_t,7}$ from an exponential distribution with mean $1/\beta$. Draw $a_{N_t,8}$ from a uniform distribution on $[t_0, t_0\Delta]$. Set $a_{N_t,9} = \infty$. Set $a_{N_t,10} = \infty$.

In the second data structure keeping track of the tree, record, that individual N_t is a branch coming off from i .

We also need to update the next person that i will infect exactly as in the previous case. Draw w from a distribution that exponential with mean $1/\beta$ if $a_{i2} = \text{Free}$ or mean rhosQ/beta if $a_{i2} = \text{SelfQuarantine}$, and set $a_{i7} = t_0 + w$.

Note that a check on the algorithm being done correctly is that we should always have a_{i1} equal to I or I^s and $a_{i2} \neq \text{Quarantine}$, when this updating rule has been called.

- Test (a_{i8}).

If the next event is that i is tested at t_0 , then we first see if the test result will come back positive or negative. If $a_{i1} \in I, I^s$, then the test is positive with probability p_{test} . Otherwise it is surely negative.

If the test is negative, then we set $a_{i4} = \text{Waiting}^-$, set $a_{i8} = t_0 + \Delta$, and $a_{i9} = t_0 + d_{\text{test}}$.

If the test is positive, then we set $a_{i4} = \text{Waiting}^+$, set $a_{i8} = \infty$, and $a_{i9} = t_0 + d_{\text{test}}$. The setting $a_{i8} = \infty$ is done because i will be permanently quarantined before any future test comes back so we simplify and avoid overwriting results by just assuming no more tests are done.

- Test result available (a_{i9})

If the next event is that i 's test result becomes available at t_0 then what we do depends on the test.

If it is negative, $a_{i4} = \text{Waiting}^-$, then we simply update a_{i4} to Known, update a_{i9} to ∞ , and change a_{i2} to Free if the person was previously in Self-Quarantine.

If it is positive, $a_{i4} = \text{Waiting}^+$, then we make several changes. a_{i2} updates to Quarantine. a_{i4} updates to Known. a_{i7} updates to ∞ to reflect that i will no longer infect anyone. a_{i8} updates to ∞ to reflect that we no longer need to test i . a_{i9} updates to ∞ as well.

In addition, a positive test by i starts contact tracing that will may update the rows corresponding to several others. Let J be the set of individuals who are either the one person who infected i or the people who have been directly infected by i . For each individual $j \in J$ we set a_{j3} to Identified with probability p_{contact} and to Unidentified with probability $1 - p_{\text{contact}}$. If j will be identified we draw a random variable x from a lognormal distribution with mean μ_{contact} and standard deviation σ_{contact} and set the time at which the contact will be made, a_{j10} , to $t_0 + x$ unless a_{j10} is already smaller than this (and $a_{j3} = \text{Identified}$) in which case we don't update a_{j10} . (The variance here is just to keep all events happening at different times so we don't have to update multiple rows at once.

For individuals who will be unidentified we can just set a_{j10} to ∞ to streamline the calculations.

- Contact traced to positive case (a_{i10})

If the next event is that i is connected to a confirmed positive case at t_0 , then we update several things about i . First, a_{i2} is updated to SelfQuarantine. IF the person is not already going to receive a positive test from a test that has already been taken, i.e., if $a_{i4} \neq \text{Waiting}^+$, then we set the time of the test result triggered by the contact tracing to $a_{i9} = t_0 + d_{\text{test}}$, and set the result of the test a_{i3} to Waiting^- or Waiting^+ , choosing the positive outcome with probability p_{test} if s_{i1} is equal to I or I^s .

If $a_{i3} \in \{I, I^s\}$, we also update the time at which i will infect someone else to $t_0 + x$, where x is a random draw from an exponential distribution with mean ρ_{SQ}/β .

We set a_{i10} to ∞ to indicate that we are not waiting for another contact trace to occur. (Note that this assumes that if contact tracing finds someone who is awaiting a test result, then the person is not retested.)

This is a long