

The Travelling Disease

Introduction to Computer Simulations

Abhi Agarwal & Philip Ottesen

1 Introduction

In this simulation, we consider the stochastic evolution of a disease across an isolated system of four cities connected by roads. Note that the system represents an undirected, connected graph. Each city i will contain the disease \mathcal{D} to a differing degree, where \mathcal{D} is assumed to be a non-lethal disease spread by contact. To model four unique cities with environmental variables that realistically differ such as sanitation levels and availability to medical treatment, we let the resultant infectivity a_i of the disease and growth rate b_i be unique. To introduce travel, we define travel rates to be the probability per unit time of an individual travelling from city i to j . We examine the behaviour of the system as \mathcal{D} evolves, whilst noting the growth of the disease in individual cities, through the use of pie charts. The fraction of susceptible, infected, and recovered citizens of each city with respect to the total population is plotted, with the corresponding proportion for the global population charted to present the state of the isolated system at any time-step.

First, we consider a stochastic model with static travel rates. These rates are systematically varied to consider disparities in the evolution of \mathcal{D} between cities with high immigrant rates versus low immigrant rates. To verify com, we show that populations will tend to migrate towards cities with high incoming and low outgoing travel rates. We also consider configurations with an epidemicity e given by the number of cities where $a_i > b_i$ holds. $a_i < b_i$ is then the case for $n - e$ cities, where n is the number of cities. An edge-case where the reproduction number R_0 of all cities is 1. Here, travel rates are varied to study what range of approximate range of rates maximize the spread of \mathcal{D} , and whether an upper-bound to the maximum number of cases over the duration of \mathcal{D} exists. The model is extended to incorporate dynamic immigration rates, with probability per unit time proportional to $\frac{I_i}{N_i}$. The resultant traffic flows will exhibit a bias towards cities with less severe outbreaks, modelling the movement of a population in an epidemic. A citizen may not have access to the reproduction number of the disease, but media outlets may provide an estimate of the above proportion, resulting in biased traffic flows. Lastly, we combine dynamic rates with an ability to quarantine incoming and outgoing travel.

2 Theory

We present our modifications to the Kermack & McKendrick compartmental SIR model. Let $F_{\alpha,i,j}$ be the matrix of probabilities per unit time for an individual in α to travel from i to j , where α is a class from the SIR model. Note that diagonal entries in F are zero, since the probability of remaining in a city at a given time-step is implicitly defined as the complement of leaving the city. At each time-step, we aggregate any traffic leaving i to j and entering i from j . First, we show the deterministic SIR model with traffic rates.

$$\frac{dS_i}{dt} = -a \frac{I_i}{N_i} S_i + \sum_{j=1, j \neq i}^4 S_j F_{S,j,i} - S_i \sum_{j=1, j \neq i}^4 F_{S,i,j} \quad (1)$$

$$\frac{dI_i}{dt} = a \frac{I_i}{N_i} S_i - b I_i + \sum_{j=1, j \neq i}^4 I_j F_{I,j,i} - I_i \sum_{j=1, j \neq i}^4 F_{I,i,j} \quad (2)$$

$$\frac{dR_i}{dt} = b I_i + \sum_{j=1, j \neq i}^4 R_j F_{R,j,i} - R_i \sum_{j=1, j \neq i}^4 F_{R,i,j} \quad (3)$$

Note that the total population size for the isolated system is N_t , where $\sum_{i=1}^4 N_i = N_t$. We discretize time with epochs $= 0, 1, 2, \dots$. The unit of the epoch in this model is the day. Let $P(I) = \frac{a_i I_i}{N_i} dt$ and $P(R) = b_i dt$. Let p denote a uniformly distributed random variable. Similar to the Tuckwell & Williams model¹, we define the following stochastic processes.

$$I^i(p, t) = \begin{cases} 1 & : p < P(I) \\ 0 & : p \geq P(I) \end{cases}$$

$I^i(p, t)$ gives the state of individual i at time t . If $I^i(p, t) = 1$, that individual is infected. Hence, the total number of infected individuals is given by $I(i, t) = \sum_{k=1}^{N_i} I^k(p, t)$ for $t \geq 0$. Similarly,

$$R^i(p, t) = \begin{cases} 1 & : p < P(R) \\ 0 & : p \geq P(R) \end{cases}$$

$R(i, t) = \sum_{k=1}^{N_i} R^k(p, t)$ gives the number of newly recovered individuals at epoch t for $t \geq 0$.

Aggregation of incoming and outgoing travel is probabilistic. To compute the flow of individuals in an (i, j) pair at epoch t , we use random processes similar to those above. Unless stated, we assume that infected individuals travel at different rates from susceptible or recovered individuals, which models a symptom of \mathcal{D} where the disease has some influence on the ability to travel. This will also simplify an interesting case where we quarantine cities, decreasing the probabilities per unit time for infectious migration of an individual. Hence, we use two random processes, $F^{SR,i,j}$ and $F^{I,i,j}$, to aid in aggregating migration between cities. In general,

¹ <http://personal-homepages.mis.mpg.de/tuckwell/tuckwell2007mathbiosciwilliams.pdf>

$$F^{i,j}(p, t, \beta) = \begin{cases} 1 & : p < dtF_{\beta,i,j} \\ 0 & : p \geq dtF_{\beta,i,j} \end{cases}$$

$F(\beta, i, j, t) = \sum_{k=1}^{\beta_i} F^{i,j}(p, t, \beta)$ gives the number of individuals from the class β who have left city i to city j . Hence, for each time-step of the simulation, $\beta(i, t) = \beta(i, t) - F(\beta, i, j, t) + \sum_{j=1, j \neq i}^4 F(\beta, j, i, t)$. To constrain the amount of travelling individuals at any given time-step, we restrict the probability $F_{\alpha,i,j}dt \ll 1$. We use the following heuristic: at each time-step t , we aggregate incoming and outgoing traffic for the ordered tuple (i, j) , where $i \leq j$. Hence, each city is given its round of traffic aggregation at each time-step, and we do not consider (j, i) since the tuple (i, j) have been processed at an earlier iteration.

The stochastic SIR model is activated when fluctuations in populations due to travel become somewhat stable. Empirically, this occurred after at most $\frac{t}{8}$ simulated days in our set of trials, where t is the number of days to simulate. This value can be computed analytically, but this solution is not considered here. This allows populations to settle before the SIR model in each city is introduced, and allows for a probable extension of the model to incorporate dynamic travel rates.

To implement dynamic travel rates, consider a city i . If an individual in i had access to information on the severity of the disease (in this model, the parameter $\frac{I_i}{N_i}$), it is realistic that each individual will attempt to make an optimal decision about their travel destination. For example, visiting cities with high concentrations of infected people may increase the likelihood of becoming infected. In our extended model using dynamic travel rates, we assume that an individual will bias their travel based on the statement $I_i > S_i + R_i$ for a given day. If this statement is true, travel from i to j becomes $F_{\beta,i,j} = \frac{I_i}{N_i}$. Similarly, $F_{\beta,j,i} = 1 - F_{\beta,i,j}$ in this case, since an individual in city j will bias his decision to leave j for i . The two probabilities become dependent in this definition, whereas previously they were not. On the other hand, if $I_i < S_i + R_i$, travel rates resume their static values for that day.

Symbol	Definition
e	Number of cities with $a > b$
k	Multiplier for rates
d	Duration of disease in days
n	Number of cities with maximum incoming travel rates
τ	Day on which infection peaked
$F_{\alpha,i,j}$	Travel rates in class α from i to j
N_t	Total number of individuals in the network.

Table 1: Table of new variables.

3 Results

Unless stated, initial parameters can be found in their corresponding appendices, with comments.

3.1 Static Travel Rates

In this experiment, we study the behaviour of the system as a function of immigration rates, with all other parameters constant. We plot the resultant changes in susceptible, infected, and recovered individuals per city over time, and discuss differing trends.

To evaluate the computational accuracy of stochastic travel in this model, we will first observe the distribution of the citizenry over time with n cities $X = X_1, \dots, X_n$ whose travel rates F_{α,i,X_k} are a maximum and $F_{\alpha,X_k,j}$ are a minimum. Let $1 \leq n \leq 4$. Assuming that traffic aggregation does not skip or miscount travel for some city, we should be able to predict the distribution of the citizenry across cities. That is, for $n = 1$, one city should have a notable maximum population. We compute the peak of infection on a given day, and record the day-number as τ .

The values used for F_{α,i,X_k} and $F_{\alpha,X_k,j}$ are shown below for $n = 1$. Since the first four cases consider computational accuracy, we make no distinction between travel rates for susceptible, infected, and recovered individuals. We also allow travel rates to be equal for the cities in X . The distinction made between $F_{SR,i,j}$ and $F_{I,i,j}$ in *Theory* will be studied in later trials.

```
TravelSR =  
  
      0      0.0500      0.0500      0.0500  
0.1000      0      0.1000      0.1000  
0.1000      0.1000      0      0.1000  
0.1000      0.1000      0.1000      0  
  
>> TravelI  
  
TravelI =  
  
      0      0.0500      0.0500      0.0500  
0.1000      0      0.1000      0.1000  
0.1000      0.1000      0      0.1000  
0.1000      0.1000      0.1000      0
```

Figure 1: Matrix showing travel rates between cities

3.1.1 $n = 1$

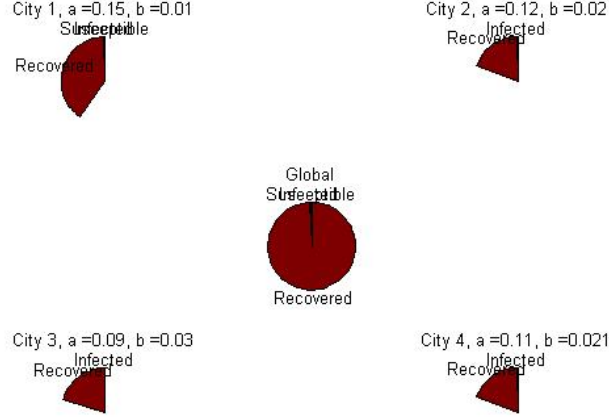


Figure 2: Pie chart for the simulation showing the proportion of infected/recovered population at a particular timestep

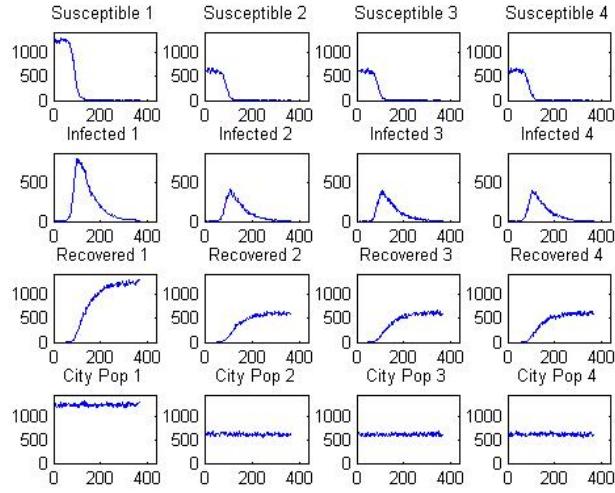


Figure 3: Graph showing the variation of the population, recovery, infected, and susceptible population

To produce the above results, we let $F_{SR=I,1,j} = 0.05$ and $F_{SR=I,j,1} = 0.1$. Note that 1 refers here to the city number, not n . Hence, it is most probable that city 1 will aggregate more incoming traffic than other cities. Indeed, the upper-left pie chart representing city 1 has accumulated a higher population

than other cities. Also, from figure 2, the populations of the three smaller cities are approximately equal. However, as expected, the fluctuations in populations are not equal, as shown by the population-plots for cities 2 and 3. Although the probabilities of an individual travelling to neighboring cities are equal in this case, the stochastic nature of traffic aggregation enforces that these migration patterns be distinct.

Note that in these cases, $a_i > b_i, \forall i = 1, \dots, 4$, which indicates that \mathcal{D} is an epidemic. The impact of a and b on the system is studied in a later section. Here, we record that the time taken for each city to reach a peak in infected population. For city 1, using the calculation shown below, we find a maximum after 126 days of infection. However, as stated in *Theory*, the infection is not introduced until after $\frac{t}{8} \approx 45$ days. Hence, $\tau_1 = 81$. Cities 2, 3, 4 gives peaks at 84, 89, 77 days respectively. This gives $\bar{\tau} = 82.75$, with a standard deviation $\sigma = 5.06$.

The method used to compute maxima is given below.

```
>> [c, i] = max(I_save(1, :))

c =

    810

i =

    126
```

Figure 4: Method used to compute maxima

3.1.2 $n = 2$

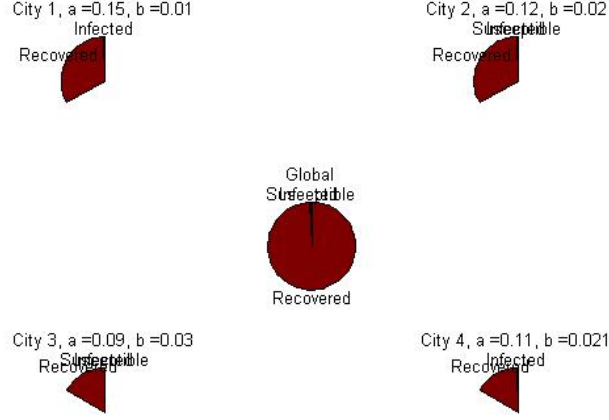


Figure 5: Pie chart for the simulation showing the proportion of infected/recovered population at a particular timestep

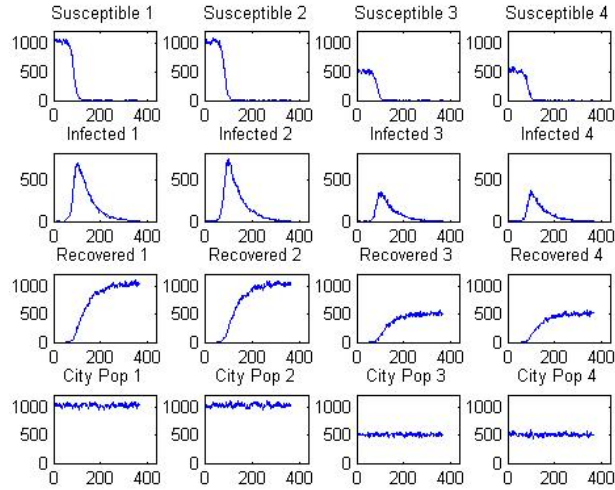


Figure 6: Graph showing the variation of the population, recovery, infected, and susceptible population

Consistent the results for our trial with $n = 1$, we observe two cities with a high population and two with a small population. We find that $\tau_1 = 56$, $\tau_2 = 59$, $\tau_3 = 54$, $\tau_4 = 63$, yielding $\bar{\tau} = 58$ with $\sigma = 3.92$. For $n = 1, 2$, we see that the peak intensity of the infection is reached in approximately the same week.

3.1.3 $n = 3$

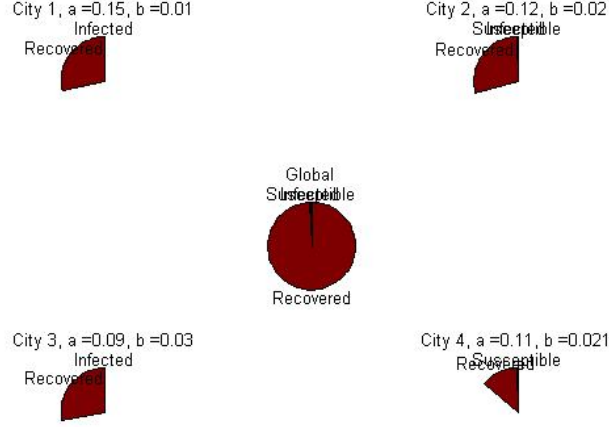


Figure 7: Pie chart for the simulation showing the proportion of infected/recovered population at a particular timestep

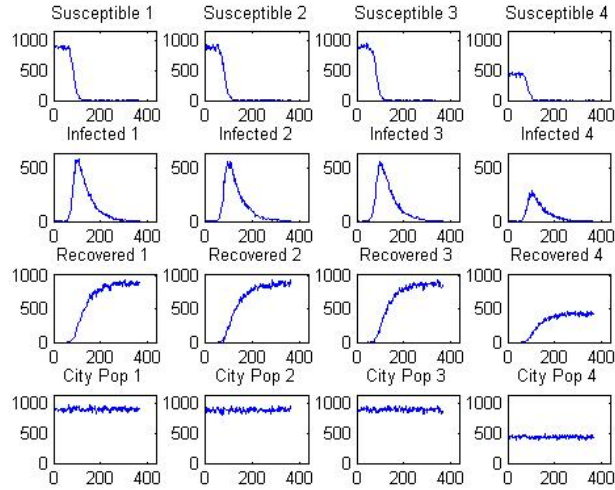


Figure 8: Graph showing the variation of the population, recovery, infected, and susceptible population

From figure 7, it is apparent that the population of the system is concentrated in cities $1, \dots, 3$. The populations of these cities tends to be more volatile than the minor city. Note the relative amplitude of oscillations in cities $1, 2, 3$ compared to city 4. Although it is more likely for an individual in 4 to travel to

1, 2, 3, the major cities have almost twice as many citizens as 1. Recall that the case studied here allows travel from 4 to its neighbors with a probability of 0.1, with a probability of travel to 4 of 0.05 for all other cities. The difference in probabilities is not sufficient to prevent travel to 1 entirely. We observe that although $F_{\alpha,j,4} = 0.05$, we have that $N_j \gg N_4$ such that the expected number of people $F_{\alpha,j,4}N_j$ travelling in a day is greater than $F_{\alpha,4,j}N_i$. Hence, we expect fluctuations in populations of the major cities to be more prominent than minor cities. We saw this for $n = 1, 2$ in figures 3 and 6. For $n = 4$, the previous assertion is no longer useful since N_j tends towards N_i as the population is no longer concentrated and we have equal travel rates. However, this may not hold for larger travel rates. As travel rates approach 0, we conjecture that the system tends towards deterministic behaviour. This will be investigated in the next section.

For $n = 3$, we have $\tau_1 = 64$, $\tau_2 = 57$, $\tau_3 = 62$, $\tau_4 = 55$ giving $\bar{\tau} = 59.5$ with $\sigma = 4.20$.

3.1.4 $n = 4$

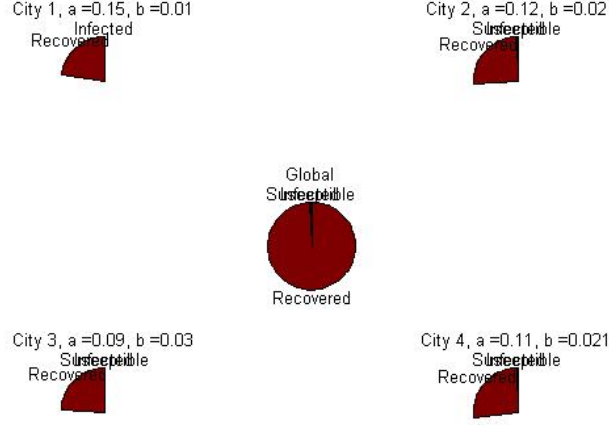


Figure 9: Pie chart for the simulation showing the proportion of infected/recovered population at a particular timestep

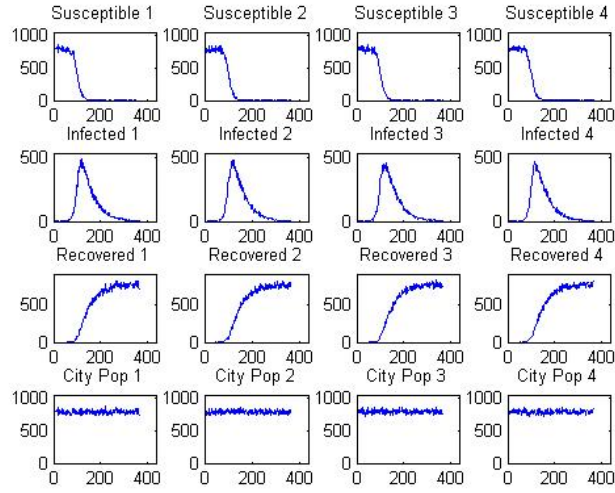


Figure 10: Graph showing the variation of the population, recovery, infected, and susceptible population

Indeed, following our arguments in the previous case, the populations across the network do not become concentrated at the conclusion of the program. From figure 9, the respective pi-charts are similar in size when the maximum travel rate between any two cities is 0.1.

We have $\tau_1 = 71$, $\tau_2 = 68$, $\tau_3 = 70$, $\tau_4 = 68$ giving $\bar{\tau} = 69.25$ with $\sigma = 1.50$. Our observation in the previous sections that the peak intensity of \mathcal{D} reached within a week for each city is consistent for $n = 1, 2, 3, 4$ with the given travel rates.

In *Theory*, we made a distinction between $F_{SR,i,j}$ and $F_{I,i,j}$. For the next three trials, we evaluate the result of this distinction. We now observe how inequalities between $F_{SR,i,j}$ and $F_{I,i,j}$ affect the spread of \mathcal{D} across the network. We can here note that we're starting with an epidemic in each city, so the following three trials will take place in a network with $n = 4$.

3.1.5 $F_{SR,i,j} = F_{I,i,j}$

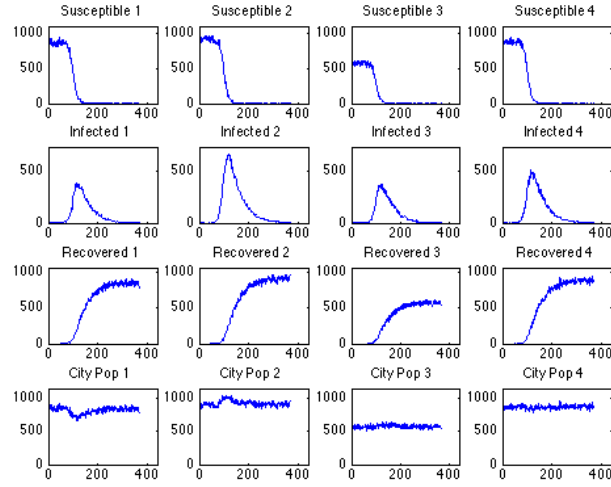


Figure 11: Graph showing the variation of the population, recovery, infected, and susceptible population

In this case for all the cities we have set all the traveling rates for the susceptible, infected, and recovered population to be the same. The statement basically means that there was an equal probability per unit time for a diseased person to come in to a city as a recovered or susceptible person, so we are assuming that a healthy and a sick person are equally as likely to travel. We also should note that in this example, each city does not have the same travel rates as one another. Each city is not taking in the same amount of individuals that are leaving.

From the figures above we can see that for each city the structure of each sub-graph is very similar, and that each city experiences levels of infection, and recovery in a very similar way. There is a rise in the number of people infected as the number of susceptible individuals fall for each city, and then as infection falls there is a rise in recovery, which is clearly what we should be seeing in our model.

We can see that there is established travel as when infection rises the population of city two increases, and this is because the travel rate of city two is higher than the rest, and so it draws in more individuals as infection peaks. As city two draws more infected people, this increases the risk of being infected in city two as our infection spread is determined by the ratio of infected people over the city's population.

We will reference this model later as we develop and look at the other cases of travel rates for S, I, and R. We should note that it takes 376 days for this particular model to eradicate the disease

3.1.6 $F_{SR,i,j} > F_{I,i,j}$

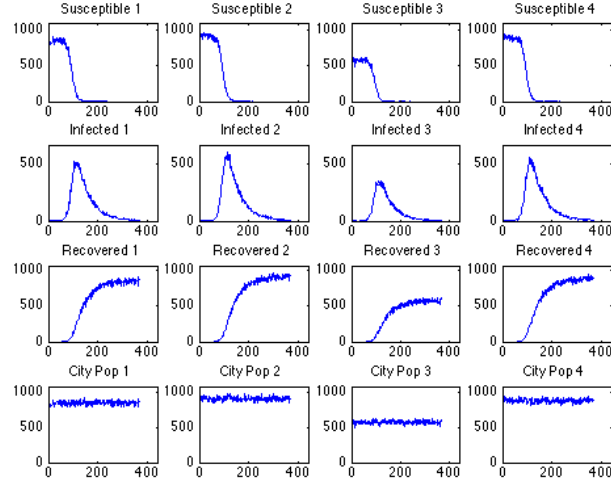


Figure 12: Graph showing the variation of the population, recovery, infected, and susceptible population

In this case we explored when for all the cities, all the traveling rates were the higher for the susceptible, and recovered population than the Infected population. This is what is depicted in the real world, but we're only considering static rates for travel, and so this would simulate as if no travel changed even after the diseases is critical.

This particular model takes 388 days to eradicate, which is interesting to look at when compared to $F_{SR,i,j} = F_{I,i,j}$. It's interesting to see that when there are more susceptible individuals traveling there is still a higher recovery date when compared as it can be a little counter intuitive at first. The data analysis and graph show that because there were a higher number of susceptible individuals traveling at the beginning they experienced different levels of infection, and different stages of infection in a city. As there are individuals traveling from low infected cities to high infected cities they have a higher chance of getting the disease, and in the beginning when the number of susceptible individuals is higher and they are traveling they are very likely to catch the disease. This peaks when the ratio of susceptible individuals to infected individuals reduces to one, and this is when the peak occurs, and as the number of susceptible people traveling is falling.

3.1.7 $F_{SR,i,j} < F_{I,i,j}$

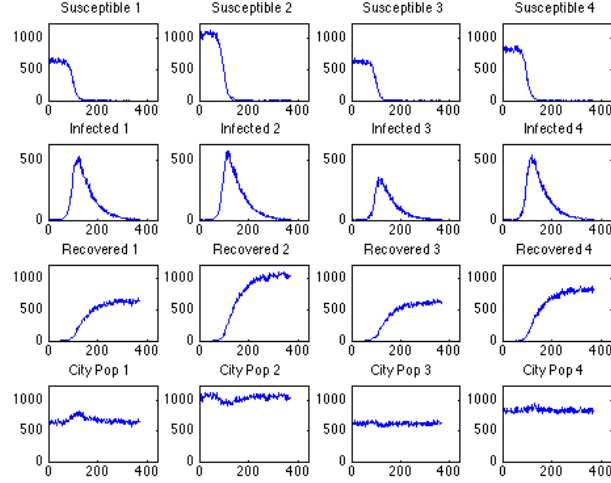


Figure 13: Graph showing the variation of the population, recovery, infected, and susceptible population

In this case we explored when for all the cities, all the traveling rates were lower for the Susceptible, and Recovered population than the Infected population. This implies that sick or unhealthy individuals are more likely to travel than recovered or healthy individuals, which is unlikely to happen but still important to study.

In this example we see the massive spike and falls in city population as infection peaks, and this is because individuals are traveling more as infection rises, and this is because as infection rises there are more infected people, and so this increases travel in general, which increases spread. There is a chain where as more people travel to highly infected cities the spread multiplies, and we can see that in the huge spikes that happen in City 1 and City 2 as we reach peak infection.

This particular disease takes 424 days to eradicate, which is more than those proposed above. This should be the case as individuals are traveling through cities with different $\frac{1}{b}$ or recovery rates, and they experience recovery in different ways. This is something we can see in the real world as well, if individuals were more likely to travel while they were sick then the recovery would take longer. This case is a little less unlikely to happen in the real world as governments usually intervene and wouldn't allow this to happen, and also the fact that weakness increases in sickness so not everyone is physically able to travel.

3.1.8 Observations on $F_{SR,i,j}$ and $F_{I,i,j}$

It is very interesting to see here that there is a somewhat trend between the travel rates, and the duration of the disease. There is an increased duration of the disease if we let more sick individuals travel, which is intuitive to understand,

but even with static travel rates solidifies our understanding of this particular model. We will be looking more at this comparison when we study Dynamic Rates and develop our understanding of Static Rates further.

3.2 Reproduction number $R_{0,i}$

In this section, we study the spread of \mathcal{D} under varying reproduction numbers $R_{0,i}$. First, we examine the case where e cities have $R_{0,i} > 1$. We will only examine cases where the network does not initially experience an epidemic. Hence, a case with $e = 4$ is ignored. Interesting cases are $e = 1, \dots, 3$.

To quantify the severity of \mathcal{D} , we plot $\frac{I_t}{N_t}$ over time.

Since our model limits contacts to social encounters, i.e. no airborne spread, we maintain $5 \leq R_{0,i} \leq 7$ for each city with $a > b$, similar to smallpox².

Note that the simulated time is 2 years, with $dt = 1$ day. The outbreak of \mathcal{D} therefore occurs after $\frac{730}{2} \approx 91$ days of idle travel. The default travel rates from appendix A are used for the following trials.

3.2.1 $e = 1$

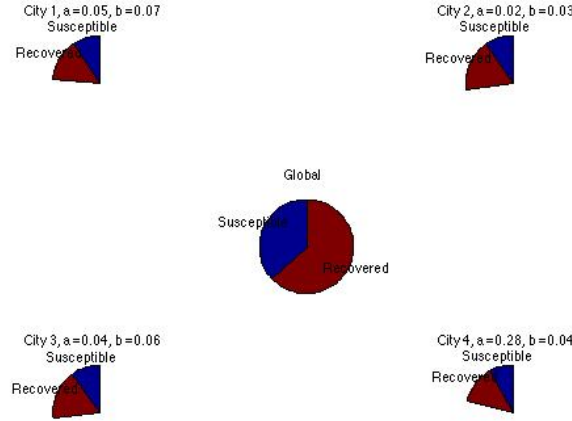


Figure 14: Pie chart for the simulation showing the proportion of infected/recovered population at a particular timestep

²<http://www.bt.cdc.gov/agent/smallpox/training/overview/pdf/eradicationhistory.pdf>

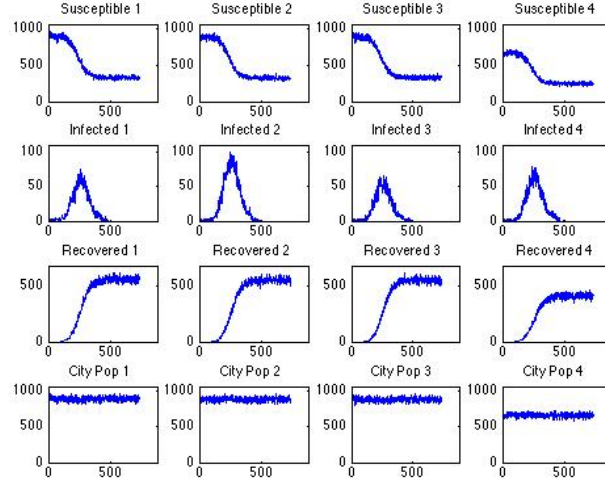


Figure 15: Graph showing the variation of the population, recovery, infected, and susceptible population

With the values of a_i and b_i shown in figure 17, \mathcal{D} only reached approx $\frac{5}{8}$ of the total population. Figure 18 shows that the infection had a duration of approx. $d = 500 - 91 = 409$ days of infection. Interestingly, given that $\max\{b_i\} \forall i = 1, \dots, 4 = 0.07$, the maximum time taken for an individual to recover from \mathcal{D} is $\frac{1}{0.07} = 7$ days, the disease survived 409 days in the network. However, $e = 1$ was not sufficient to infect all members of the population. Also note that $\tau = 260$.

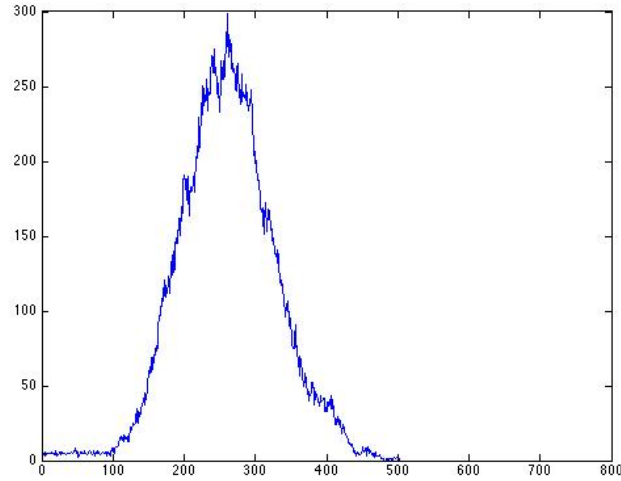


Figure 16: Graph showing the variation peak of the infection

3.2.2 $e = 2$

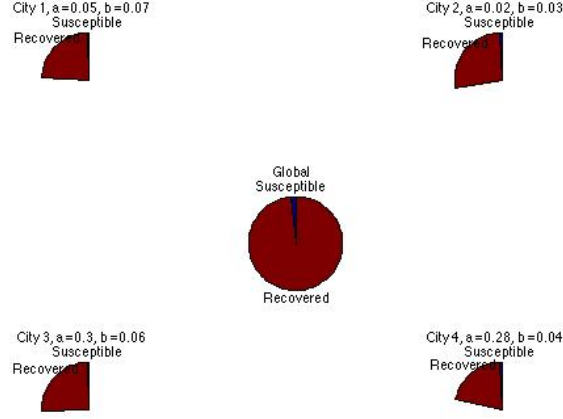


Figure 17: Pie chart for the simulation showing the proportion of infected/recovered population at a particular timestep

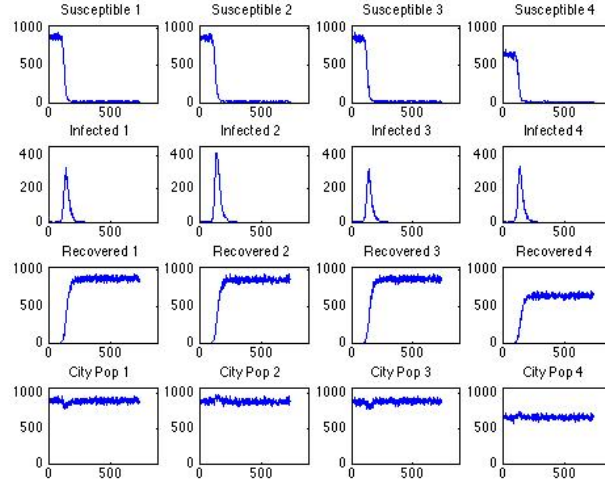


Figure 18: Graph showing the variation of the population, recovery, infected, and susceptible population

Compared to figure 17, we see an immediate change in the behaviour of \mathcal{D} . It spread through the network significantly faster than $e = 1$, as expected. We have two cities with conditions for an epidemic contributing to τ . We find that $\tau \approx 52$, with a peak of 1400 infected or a ratio to the total population of

$\frac{1400}{N_i} = \frac{1400}{3100} = 0.45$. Although the infection peaked earlier than $e = 1$, and with a larger maximum, the number of infected individuals across the network reached 0 after $d = 300 - 91 = 209$ days of infection. Hence, the disease lasted half as long as $e = 1$. There appears to be an inverse relationship between e and the duration d and day of maximum infected individuals τ , while an earlier τ implies a shorter duration d . The expected number of individuals moving from I to R for any given city for each time-step is proportional to the number of infected people in that city, since $R(i, t) = \sum_{k=1}^{N_i} R^i(p, t)$. Therefore, if \mathcal{D} saturates the network with infected people early on, $R(i, t)$ will be large early on, decreasing with t . For $e = 1$, we observed a small pocket of the network infected over the period of the simulation, and hence, $R(i, t)$ is sufficiently small to allow a small proportion of infected people to move through the network. Hence, this small infected group faces less resistance when moving through the network compared to the growing concentration seen in $e = 2$. This corresponds to the observation that every individual can only be infected and can only recover once, such that the critical $R(i, t)$ marks a significant number of infected individuals as recovered for each time-step. An interesting complication of this model would be to neglect immunity from \mathcal{D} such that there exists a reverse process from R to I .

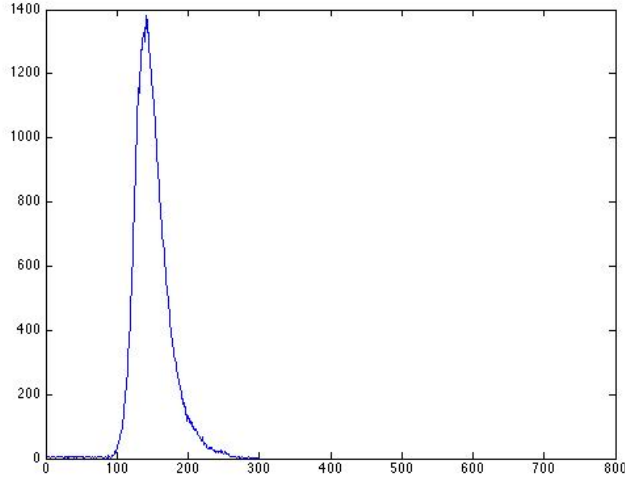


Figure 19: Graph showing the variation peak of the infection

3.2.3 $e = 3$

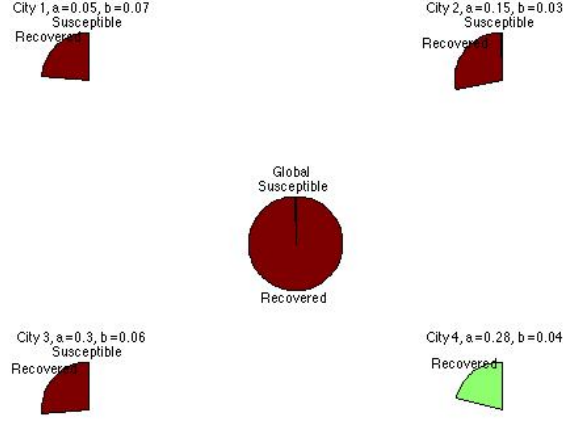


Figure 20: Pie chart for the simulation showing the proportion of infected/recovered population at a particular timestep

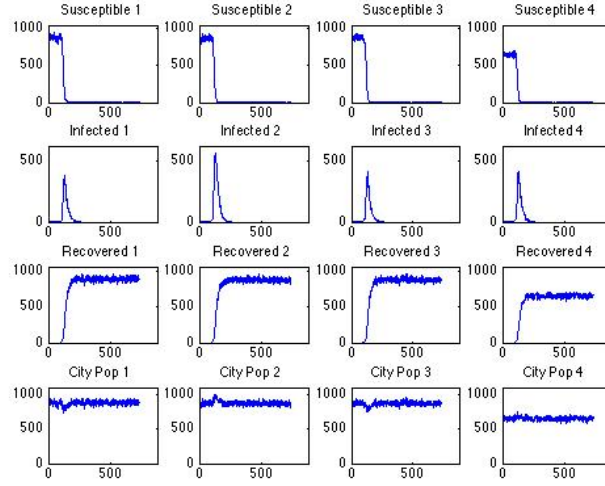


Figure 21: Graph showing the variation of the population, recovery, infected, and susceptible population

Confirming our results from $e = 2$, we find a smaller $\tau = 130 - 91 = 39$ with a higher peak of 1635 infected individuals. The duration $d = 262 - 91 = 171$ days, and again we see that as e increases, the duration d and τ are left-shifted.

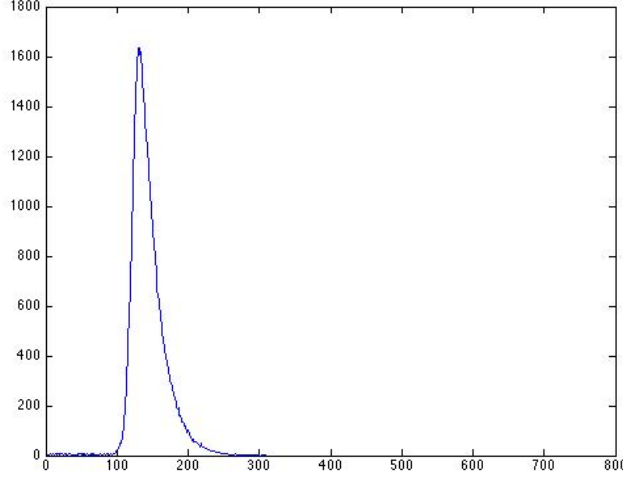


Figure 22: Graph showing the variation peak of the infection

3.3 $R_{i,0} = 1$

In a deterministic or stochastic SIR model without travel, with a reproduction number $R_{0,i} = 1 \forall i = 1, \dots, 4$, the disease will not spread throughout a given city. This can be shown using our model by setting all travel rates to 0. From a previous study on the number of cities with $R_{0,i} > 1$, we showed that pockets of infected individuals moving through the network increase the duration of \mathcal{D} . An infected individual will recover slower in cities with smaller b . The travel of infected individuals through the network is directly related to the duration of the disease, which suggests that there might be optimal travel rates that increase the total number of individuals who have contracted \mathcal{D} .

However, once travel is introduced, it is possible to propagate the small fraction of infected people in each city such that \mathcal{D} infects a significant proportion of the total population. We show this by considering the proportion $\frac{I_t}{N_t}$, where I_t gives the total number of infected people.

To vary travel rates systematically, we introduce a multiplier $0.003k$, where k is some positive number. The multiplier is used to increment all travel rates, with varying k . We posit that for large k , i.e. a busy network since travel rates are high, we should observe a well-mixed population such that the dominant spread of \mathcal{D} is through the SIR model. We can see this by comparing high travel rates with the 1-city case. We have chosen rates such that a low k will not affect the expected outcome of the disease, i.e. very short duration when $R_{i,0} = 1$. Hence, there appears to be some optimal k such that the number of infected people at any given time reaches a maximum.

Below, we see the travel rates used for this experiment. The number of individuals in each city has not changed, but $dt = 0.5$ days. We let $k \leq 50$ such that the restriction $F_{\alpha,i,j} * dt < 1$ holds for this set of trials. Unlike $k = 50$, we give an extended analysis for the case $k = 11$ since $F_{\alpha,i,j} * dt \ll 1$ and

$\max\{F_{\alpha,i,j} * dt\} = Z$. However, since we attempt to show an upper-bound on the range of optimal travel rates, we consider the extreme $k = 100$, although $\max\{F_{\alpha,i,j} * dt\} = 0.27$.

```
>> TravelSR
TravelSR =
    0    0.4000    0.5200    0.3900
    0.4900    0    0.4000    0.4000
    0.5100    0.4500    0    0.5000
    0.4000    0.5000    0.3300    0

>> TravelI
TravelI =
    0    0.1830    0.2470    0.1430
    0.2230    0    0.1660    0.1540
    0.3230    0.1830    0    0.2330
    0.1630    0.2330    0.2830    0

>> TravelII
TravelII =
    0    0.0830    0.1330    0.1230
    0.0330    0.1310    0.1190    0.1280
    0.1460    0.1630    0    0.1730
    0.1610    0.1300    0.1580    0

>> dt
dt =
    0.5000
```

(a) Initial travel rates.

(b) Incremented travel rates for $k = 100$.
Note that the probability for travel is $F_{\alpha,i,j}dt$.

Figure 23

3.3.1 $k = 5$

For low k , we posited that this particular system will not exhibit a significant spread of \mathcal{D} . Recall that each travel rate from figure 26 has been incremented by $0.003(5) = 0.015$.

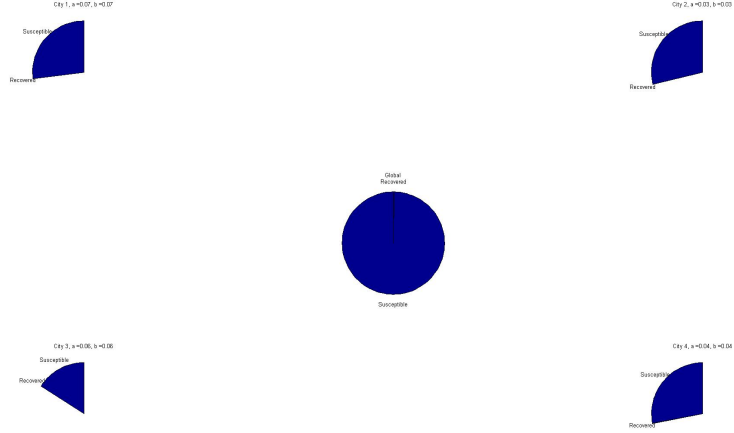


Figure 24: $k = 5$.

Indeed, the proportion of recovered individuals is negligible, and equivalent to a case with no travel between the cities. Line 81 of appendix B was used to show the number of newly infected individuals. For $k = 5$, we found a negligible 8 new cases of \mathcal{D} over the period of two years.

3.3.2 $k = 50$

We found that the disease spanned a comparatively large area of the global population chart for $50 \leq k$ in contrast to $k = 5$. In particular, $k = 50$ gave $I_t \geq \frac{N_t}{4}$, where I_t is the global number of infected individuals.

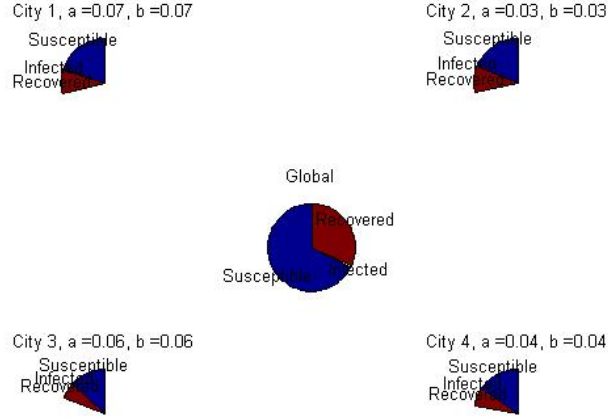


Figure 25: $k = 50$.

From figure 28, the spread of \mathcal{D} for $k = 50$ clearly dominates over $k = 5$. The trial can be repeated using code from appendix B, and one can test whether the maximum at $k = 50$ is a statistically significant finding. However, since k is an arbitrary measure of the optimal set of travel rates, we will focus on testing our hypothesis rather than computing an exact value for the optimal k .

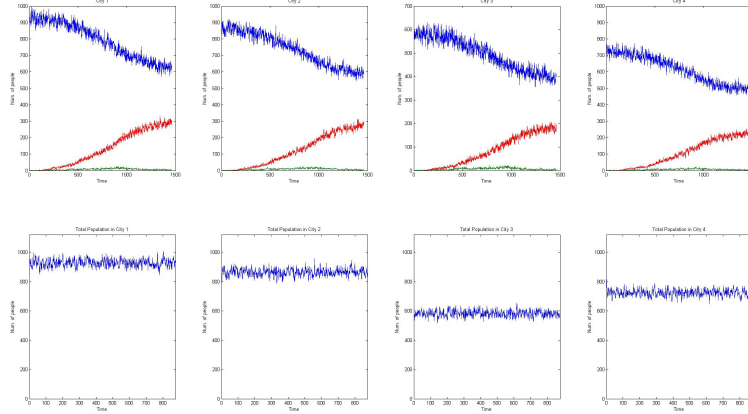


Figure 26: Plot of population distribution over time. $B = S, R = R, G = I$.

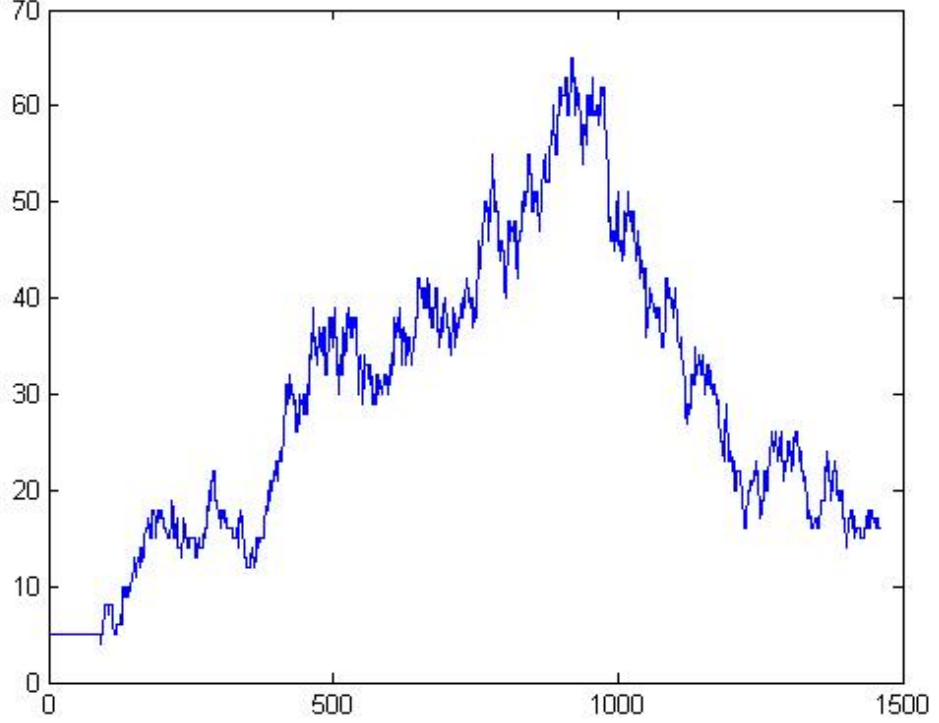


Figure 27: Number of globally infected individuals on any half-day.

From the figure above, we see that the susceptible and recovered lines are close to intersecting. $\frac{dS}{dt}$ and $\frac{dR}{dt}$, excluding stochastic travel, converge to 0 near the end of the simulation. Since the x-axis is in half-days, this corresponds to roughly $\frac{1500}{2} = 750$ days. Hence, the number of individuals who have been infected and eventually recovered is nearly half the global population. Due to stochastic noise, this has not always been the case. In our runs, we have found that $k \geq 50$ tends to produce $\frac{1}{4}$ recovered individuals. Note the small values of infectivity and growth rate from figure 28, and recall that a similar observation to the following was made earlier. Although the recovery period for an individual is at most 1 week (city 1), we have a duration of approx. $d \approx 750$ days. We refrain from taking the difference with a 91-day offset, since the infection was not entirely wiped out, 750 days is a reasonable upper-bound based on previous experiments where the disease lingers for a few weeks. Where we initially had a short-lived disease for $k = 5$, we now have a disease which is endemic in the population over the course of at least 2 years. By computing the total recovered individuals after the simulation, and incorporating any outlying individuals still infected, we obtain a total number of infected individuals of 999 or 32% of the global population for figure 29. For a more precise value, one might increase the bound on simulated time, to ensure that the infection dies out by the end of the simulation. From figure 30, we find that $\tau = 918\frac{d}{2} \approx 459$ day with a peak of infection of 65 individuals. However, since the SIR model was not activated

until 91 days into the simulation, we have $\tau = 459 - 91 = 368$.

3.3.3 $k = 100$

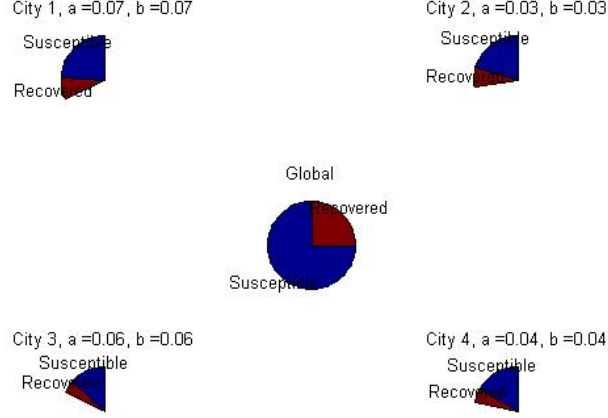


Figure 28: $k = 100$.

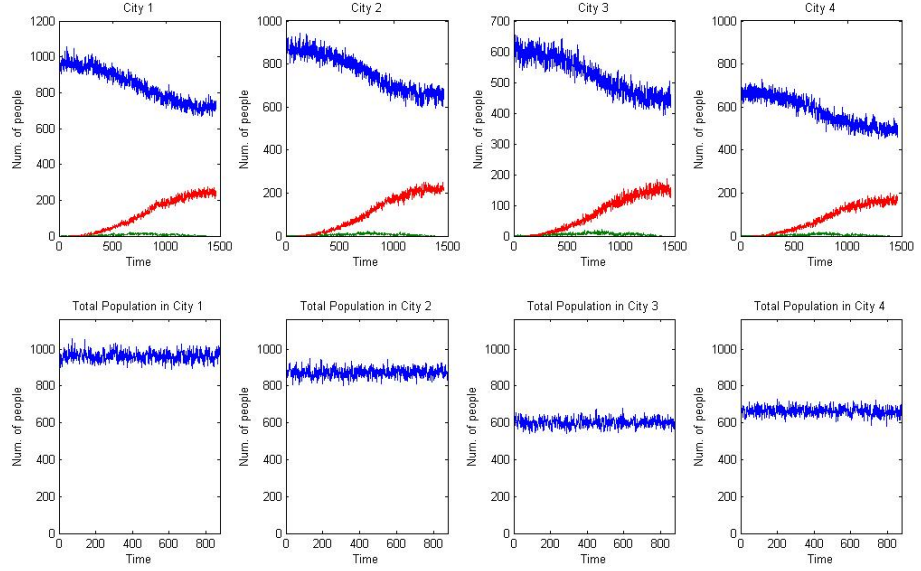


Figure 29: Plot of population distribution over time. $B = S, R = R, G = I$.

Here, we increase all travel rates by a factor of $100(0.003) = 0.3$ to consider an extreme case. Although a large increment, note that $0.3dt = 0.15$, so our

maximum travel rate $F_{\alpha,i,j}$ from our initial travel rates in figure 31 is $F_{SR,1,3} = 0.22 + 0.3 = 0.52$, with $F_{SR,1,3} * dt = 0.27 < 1$.

We see an immediate similarity between the above figures and the corresponding figures for $k = 50$. The overall newly infected individuals is reached 780 or 25% of the global population. A comparison of figures 32 and 29 will show that regardless of the large differences in travel rates from $k = 50$ to $k = 100$, while $\frac{dI}{dt}$ and $\frac{dR}{dt}$ reach 0 at different times, they tend to reach a non-epidemic steady state (since the lines never intersect). This suggests that our intuition of an upper-bound to an optimal k is consistent with results. A plausible reason for this is that as travel increases, the populations become well-mixed, and hence the behaviour of \mathcal{D} will eventually converge to that of a single city SIR simulation with $R_{0,i} = 1$ with negligible new infections. Furthermore, note from figure 33 that the number of infected individuals in each city peaks earlier than figure 30. As we deduced in the subsection on epidemicity $e = 2$, an earlier τ implies a shorter duration. Indeed, the duration of \mathcal{D} is approx. $d = \frac{1387}{2} - 91 = 603$ days, shorter than $d_{50} = 750$ days. The peak of infection occurred on day $\tau = \frac{779}{2} - 91 = 390 - 91 = 299$, with a maximum of 55 infected citizens.

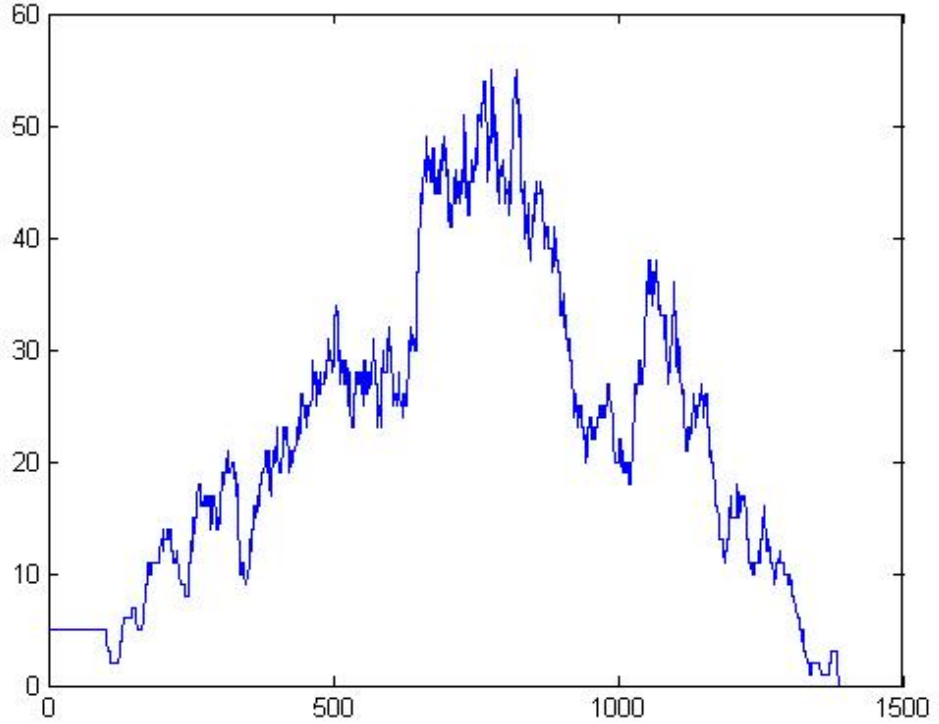


Figure 30: Number of globally infected individuals on any half-day.

3.4 $R_{4,0} > 1, e = 1$

In this experiment, we build on ideas developed in the previous section where we found optimal k producing a significant outbreak of \mathcal{D} . Here, we consider the minimum travel rates required to infect the majority of the population for $e = 1$. We will fix $e = 1$ with $a_4 > b_4$, and incrementally vary travel rates. Again, we constrain our travel rates such that the product with dt is small compared to 1.

We increment travel rates using the previous multiplier, $0.003k$, where k is a relative measure of the increase in rates across all cities. For the initial travel rates, refer to lines 31 and 32 in appendix B.

3.4.1 $k = 1$

To confirm that the above set of initial rates results are insufficient to cause an epidemic across the network, we refer to figure 34.

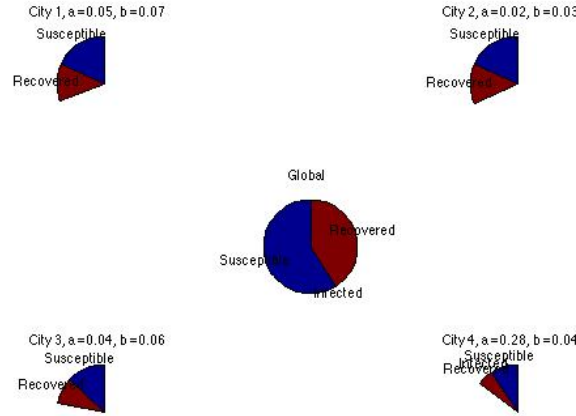


Figure 31: Pie chart for the simulation showing the proportion of infected/recovered population at a particular timestep

Indeed, as observed in the first $e = 1$ trial, we find that the system does not reach an epidemic state. Now we will investigate whether such a state is at all possible by varying travel rates alone.

3.5 $k = 5, 80$

We found that $k \leq 5$ tended to produce cases numbering between $\frac{1}{2}$ to $\frac{5}{8}$ of the global population.

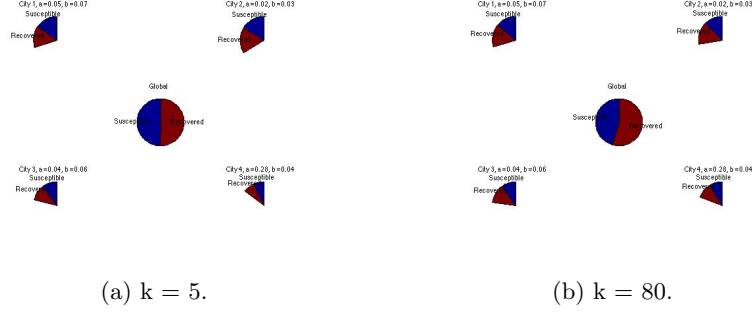


Figure 32

In the trial shown above for $k = 5$, an increment of $0.003(5) = 0.015$ in the travel rates caused approx. $\frac{1}{2}$ infections in the network. We show that this relationship is not linear. i.e. an increment of $0.003(10) = 0.03$ will not cause the entire population to become infected. In fact, the number of cases for $k = 80$ or an increase in rates by 0.24 differs from $k = 5$ by approx. $\frac{1}{8}$. Since the highest travel rate in this set of parameters is $F_{SR,4,1} = 0.076$, we have $(F_{SR,4,1} + 0.24)dt = 0.316(0.5) = 0.158 \ll 1$, so $k = 80$ is not an unreasonable test-case. We conjecture that it is not sufficient, for a general set of parameters, to cause \mathcal{D} to behave as an epidemic by increasing travel rates across the network alone. However, for $k = 5$ and $k = 80$, we see that the number of susceptible and recovered individuals converge since more than half the population becomes infected.

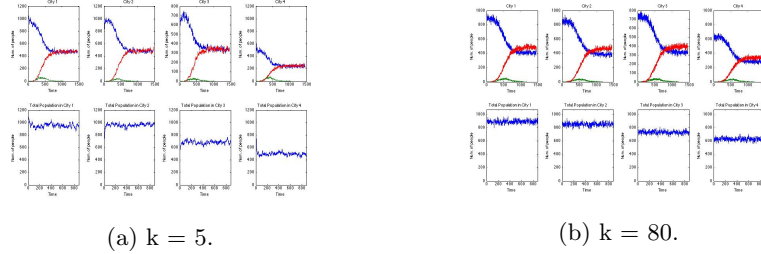


Figure 33

3.5.1 Dynamic Travel Rates

Dynamic travel rates are similar to static travel rates, but with a twist where we change the travel rates as the model evolves. When the simulation begins it runs exactly the same way as it did for the static rates until we reach a point where the number of infected people is greater than the number of susceptible/recovered people, and this is where the dynamic rates come in. We change the travel rates between cities to mirror what is going on within those cities, and the effect that occurs when panic kicks in.

Lets take an example of the panic. As the disease spreads and there is a high proportion of infected individuals to the total population of the city panic occurs as newspapers, televisions report that there is an outbreak, and in the real world this would definitely change travel patterns and so the dynamic travel rates will

try and simulate this. The travel rates for both the infected population and the susceptible/recovered population are biased by $\frac{\text{Number of infected people in the city}}{\text{Total city population}}$, and it updates the travel rates for all cities coming into city I, and leaving city I.

3.5.2 $F_{SR,i,j} = F_{I,i,j}$

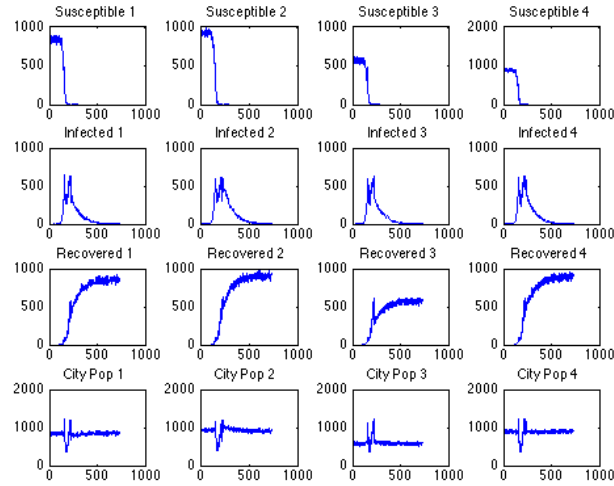


Figure 34: Graph showing the variation of the population, recovery, infected, and susceptible population

This is an interesting model to look at when comparing to dynamic rates. Firstly, the duration the duration for this particular disease is 808 days. This is important as we can clearly see that the introduction of dynamic rates as sky-rocketed the duration of the disease lasting. This is because we bias our values towards the number of individuals infected, and also because of the panic that is created. Because of the panic the values increase as susceptible people leave the city to miss the disease, and infected people leave the city to seek medical attention or for work. In this particular example we know that sick people are as likely to travel as susceptible/recovered people, and so we can see the effects of this in the graph.

3.5.3 $F_{SR,i,j} < F_{I,i,j}$

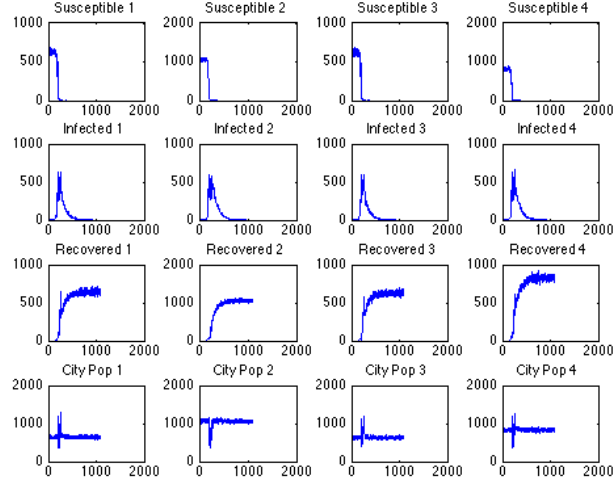


Figure 35: Graph showing the variation of the population, recovery, infected, and susceptible population

In this example we consider a scenario where we are changing the travel rates but leaning more towards more infected people traveling than more susceptible/recovered people.

The duration for this particular model is 907 days. This follows our trend of the increasing values, and this should be the highest peak in our examples as we're considering a model where we are biasing values towards the level of infection, which in turn is increasing the number of infected individuals traveling even more. This increase in travel for infected people leads to more infected people traveling as they go through a variety of recovery rates, and in this case are impacted because of the bias even further.

3.5.4 $F_{SR,i,j} > F_{I,i,j}$

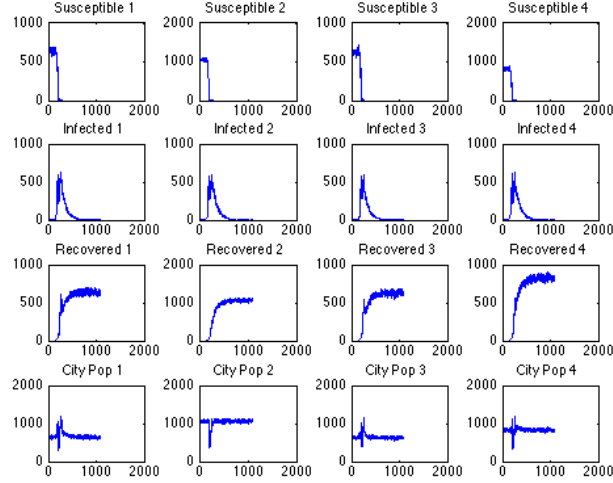


Figure 36: Graph showing the variation of the population, recovery, infected, and susceptible population

This particular model is the closest I think we possibly could get to the real world out of our examples, and a very particularly important one to study. This is close to the real world because we're able to simulate both dynamic traveling rates and the spread of disease with only a couple of limitations. The dynamic travel rates is important as it simulates an outbreak of panic, which is similar to what happens as an epidemic spreads in the real world.

We should note that this particular simulation takes approximately 631 days to eradicate the disease. This is important as this is fairly lower than the two dynamic models we have seen able, and fairly accurate. We have assumed the same difference between the susceptible/recovered traveling rates and infected travel rates as we did with $F_{SR,i,j} < F_{I,i,j}$. This is important because it establishes a trend where we are able to confirm that panic does in fact create a higher duration, and that in this example a lower travel rate for the infected population does reduce the duration, which was our hypothesis on the topic. This occurs because as we bias our results towards infection, we know that the rates will adjust so the number of susceptible/recovered individuals would be more likely to move towards cities which have the lowest infection, while the infected population would be less likely to travel when compared to them. Initially this would not matter as the model only starts after $I > SR$ for travel rates, but this would really take effect after the infection has peaked. As this model tries to adjust the travel rates for SR to be greater than the travel rates for I more individuals would have a probability of not traveling, and therefore recovering.

We also managed to simulate the effect when the individuals who are traveling travel towards cities with lower $\frac{1}{b}$ values rather than traveling to cities with higher $\frac{1}{b}$ values, which would help them recover faster. This is what actually

would happen in the real world as the travel rates would in fact change as governments, individuals, and organization educate the people about the disease. Some of the susceptible population would also move out of a highly infected city to miss the disease.

3.6 Quarantine

Quarantine is a special case where we used Dynamic Rates to simulate what governments would normally do when infection is really high, and they want to cut off travel rates. The process is basically to simulate an event that would close off travel a particular city and the rest (both incoming and outgoing).

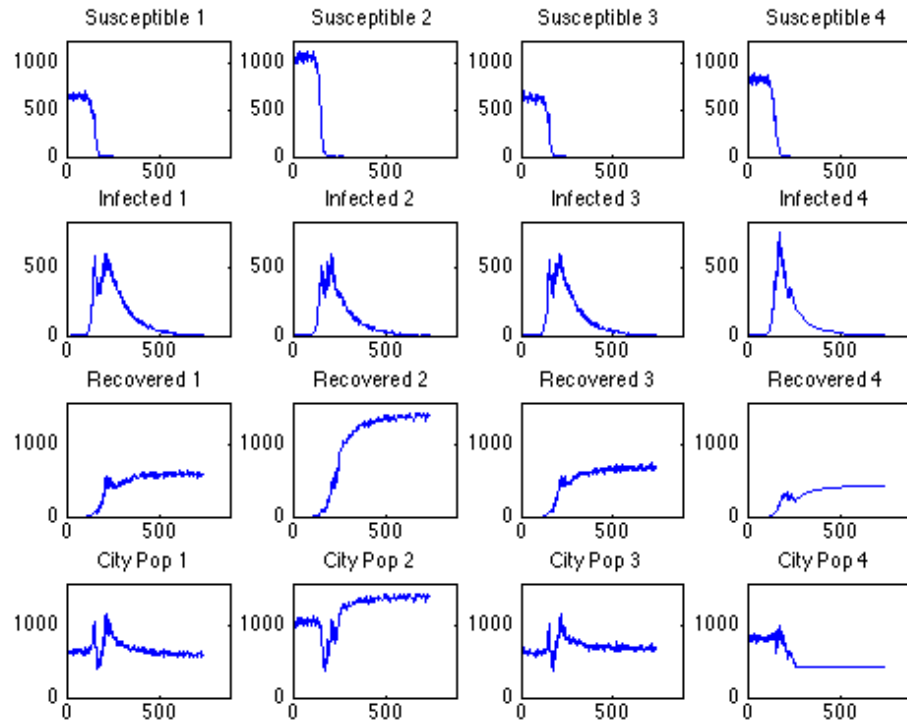


Figure 37: Graph showing the variation of the population, recovery, infected, and susceptible population

From the Figure above we are able to see the quarantine in action. If we look at City 4 we are able to see that after a period of time the travel stops and the line flattens. This is because we have set quarantine to occur when the level of infection in a city is greater than both recovery and the number of susceptible individuals. When this occurs the government would shut off all travel, and the city will recover on its own. This exactly what we see - at a point the city cuts off all its travel, and the line flattens. This is also a point where the recovery grows on its own without spikes, and infection reduces without any spikes.

This is a fair example of when the government would introduce total quarantine as individuals wouldn't be allowed to leave. The duration here is 752 for the total the whole global population, but about 710 days for City 4. So because there isn't an exchange of any more susceptible individuals or infected individuals the city is able to recover a little faster.

4 Conclusion

In conclusion, as we were attempting to model the different scenarios we found a few fundamental concepts that were apparent in all of them.

Duration of disease is inversely related to epidemicity e . We found that there exists a relationship between how many cities have the potential for an epidemic at the start and the duration of the epidemic. As we increase e in a model the duration of the epidemic increases by some constant factor. Furthermore, an earlier τ implied a shorter duration. These relationships were present in both the model for static rates, and for dynamic rates.

Travel rates and reproduction number determines behavior of disease. We found that altering the values for the travel rates and the reproduction number changed the duration of the disease, and the behavior of the disease. As we noted in the dynamic rates model changing the travel rates during the simulation changed the duration of the disease, and the fact that the ratio between the susceptible/recovered travel rates and the infected travel rates was also a big factor in determining the behavior of how the disease evolved. In addition, we also had a different reproduction number in different cities for each simulation, which allowed us notice that infection spreads differently if people recover/catch a disease in a different city, and that travel not only spreads the disease but also speeds up/slow down recovery based on the values for a and b within a city. This is important as we realize that recovery in your own city doesn't solve the issue if individuals traveling into your city have a low recovery and high infection.

When $R_0 = 1$, we found that increased travel rates could give rise to a significant spread of \mathcal{D} in the global population. The result was endemic behavior, where the duration of the disease was significantly larger than the recovery period $\frac{1}{b}$. However, we saw an upper-bound to constant used to increase travel rates. An increase in rates by a factor of 0.30 yielded an outbreak in 25% of the total population, while an increase of 0.15 produced cases in 32% of the total population. By plotting the change in susceptible, infected, and recovered individuals over time, we showed that the number of susceptible and recovered individuals tends to converge but never intersects. The upper-bound was interpreted as an inflection point wherein the global population becomes well-mixed such that the system behaves like a dominating SIR model acting on a system with no travel. We argued that since the minimum travel rates required to optimize the number of cases of \mathcal{D} across the network showed a behaviour consistent with the previous threshold study, increasing travel rates are not sufficient to cause an epidemic in general. However, whereas the epidemiology of a disease in the Kermack & McKendrick SIR model can be specified using R_0 , the epidemiology of the travelling disease cannot be classified by R_0 alone.

4.1 Future Work

4.2 Apply to SEIR, SIS, birth and deaths models

In the future we're able to introduce some new variables/alternations to the models above to make improvements or to study different variations. Firstly, we're able to study the SEIS model (E is exposed) where we consider an individual as being a carrier of the disease and removing the assumption that every individual gets the disease if they are exposed to it.

We are also able to study the SIS model where an individual goes back to becoming susceptible after recovering from the disease. Using this model we can vary other factors to measure the duration of the disease, and introduce some factor that would affect a person getting a disease.

We could also study the birth and death models as we assume that everyone at the end of the disease recovers and is able to start traveling right away. We are also assuming that over the 300-900 days there are no other individuals that are born, which in most cases wouldn't be true. Introducing death and birth would increase the complexity of this model and allow us to move another step closer to simulating a disease close to the real world.

4.3 Vary a_i and b_i over time.

Our proportion of a/b stays fixed over the time of the simulation, and as the epidemic spreads from stay one city to the other the a and b values should adjust to that spread in the other cities. If we consider a model where e (epidemicity) = 1 we aren't letting other cities grow into becoming cities with epidemics, but looking at the change in the number of people infected because more infected people travel from city 1 to the rest. As these individuals travel the values for a and b stay the same in those cities when they should be adjusting to allow for the disease to spread naturally.

4.4 Add delay in traveling between cities, and travel

We are allowing for instantaneous travel between the different cities, which would allow some error to occur. This is because an individual could potentially exist that would travel to all four different cities within a short amount of time, which wouldn't naturally happen. This causes the individual to go through too many different recovery rates and therefore could give us a few errors.

4.5 Study spread of information across networks

We are able to use this simulation to simulate the spread of information across different networks where the networks are nodes on a graph like these cities are. We have established trade between these nodes, and we are able to see how fast the information would spread across the networks similarly to the spread of diseases.

5 References

CDC, History and Epidemiology of Global Smallpox Eradication. Available at <http://www.bt.cdc.gov/agent/smallpox/training/overview/pdf/eradicationhistory.pdf>

Tuckwell, Henry C. and Ruth J. Williams, 2006, Some properties of a simple stochastic epidemic model of SIR type, *Mathematical Biosciences* 208, 76-97.
Available at <http://www.personal-homepages.mis.mpg.de/tuckwell/tuckwell2007mathbiosciwilliams.pdf>

6 Appendix A: staticRates.m

```
1 clear all;
2 clf;
3 close all;
4
5 numCities = 4;
6 time_simulated = 365 * 2; %number of days
7 clock_max = 365 * 2; %divide number of days into day
   intervals
8 dt = time_simulated / clock_max;
9
10 N_save = zeros(numCities, clock_max);
11 S_save = zeros(numCities, clock_max);
12 I_save = zeros(numCities, clock_max);
13 R_save = zeros(numCities, clock_max);
14 I_peaks = zeros(1, clock_max);
15
16 N = [1000 500 400 1200];
17 S = [999 498 399 1199];
18 I = [1 2 1 1];
19 R = [0 0 0 0];
20
21 totalPopulation = sum(N);
22
23 %Testing epidemicity e = 1,2, 3
24 a2 = [0.05 0.02 0.3 0.28]; % infectivity a = # of new
   cases per day caused by one infected person. %time
   taken to recover per person is 1/b
25 a1 = [0.05 0.02 0.04 0.28];
26 a3 = [0.05 0.15 0.3 0.28];
27 %b = [0.07 0.03 0.06 0.04];
28
29 % Default rates
30 % Just a normal travel case where we have established SR(
   all) > I(all)
31 TravelSR = [0 0.1 0.3 0.09; 0.19 0 0.10 0.10; 0.29 0.15 0
   0.20; 0.1 0.2 0.03 0];
32 TravelI = [0 0.05 0.1 0.12; 0.01 0 0.03 0.09; 0.11 0.04 0
   0.09; 0.11 0.10 0.07 0];
33 a = [0.15 0.12 0.09 0.11];
34 b = [0.01 0.01 0.01 0.01];
35
36 %F_SR > F_I, F_SR = F_I, F_SR < F_I
37 % Normal Test Case 2
38 % Case where SR/I are the same
39 TravelSR2 = [0 0.1 0.3 0.09; 0.19 0 0.10 0.10; 0.29 0.15
   0 0.20; 0.1 0.2 0.03 0];
40 TravelI2 = [0 0.1 0.3 0.09; 0.19 0 0.10 0.10; 0.29 0.15 0
```

```

0.20; 0.1 0.2 0.03 0];
41
42 % Normal Test Case 3
43 % A scenario where I(all) > SR(all) because people who
    are sick are kicked
44 % out of the city and just travel to different cities
    seeing refuge but no
45 % one gives them refuge.
46 TravelSR3 = [0 0.05 0.1 0.12; 0.01 0 0.03 0.09; 0.11 0.04
    0 0.09; 0.11 0.10 0.07 0];
47 TravelI3 = [0 0.1 0.3 0.09; 0.19 0 0.10 0.10; 0.29 0.15 0
    0.20; 0.1 0.2 0.03 0];
48
49 % Normal Test Case 4
50 % 0 are the highest leaving and lowest coming in
51 % all travel rates are equal (n = 4 same travel rate)
52 TravelSR4 = [0 0.1 0.1 0.1; 0.1 0 0.1 0.1; 0.1 0.1 0 0.1;
    0.1 0.1 0.1 0];
53 TravelI4 = [0 0.1 0.1 0.1; 0.1 0 0.1 0.1; 0.1 0.1 0 0.1;
    0.1 0.1 0.1 0];
54
55 % n = 1
56 % 1 are the highest leaving and lowest coming in
57 % 3 of them have equal travel rates
58 % high number coming in, small number leaving
59 TravelSR5 = [0 0.05 0.05 0.05; 0.1 0 0.1 0.1; 0.1 0.1 0
    0.1; 0.1 0.1 0.1 0];
60 TravelI5 = [0 0.05 0.05 0.05; 0.1 0 0.1 0.1; 0.1 0.1 0
    0.1; 0.1 0.1 0.1 0];
61
62 % n = 2
63 % 2 are the highest leaving and lowest coming in
64 % 2 of them have equal travel rates
65 TravelSR6 = [0 0.05 0.05 0.05; 0.05 0 0.05 0.05; 0.1 0.1
    0 0.1; 0.1 0.1 0.1 0];
66 TravelI6 = [0 0.05 0.05 0.05; 0.05 0 0.05 0.05; 0.1 0.1 0
    0.1; 0.1 0.1 0.1 0];
67
68 % n = 3
69 % 3 are the highest leaving and lowest coming in
70 % 1 city has the maximal travel rates
71 TravelSR7 = [0 0.05 0.05 0.05; 0.05 0 0.05 0.05; 0.05
    0.05 0 0.05; 0.1 0.1 0.1 0];
72 TravelI7 = [0 0.05 0.05 0.05; 0.05 0 0.05 0.05; 0.05 0.05
    0 0.05; 0.1 0.1 0.1 0];
73
74 startedTravel = false;
75
76 figure;
77 set(gcf, 'double', 'on');

```

```

78 subplot(3, 3, 1);
79 pie_1 = pie([S(1)/totalPopulation I(1)/totalPopulation R
              (1)/totalPopulation], {'Susceptible', 'Infected', '
              Recovered'});
80 title(strcat('City 1, a = ', num2str(a(1)), ', b = ',
              num2str(b(1))));
81 subplot(3, 3, 3);
82 pie_2 = pie([S(2)/totalPopulation I(2)/totalPopulation R
              (2)/totalPopulation], {'Susceptible', 'Infected', '
              Recovered'});
83 title(strcat('City 2, a = ', num2str(a(2)), ', b = ',
              num2str(b(2))));
84 subplot(3, 3, 7);
85 pie_3 = pie([S(3)/totalPopulation I(3)/totalPopulation R
              (3)/totalPopulation], {'Susceptible', 'Infected', '
              Recovered'});
86 title(strcat('City 3, a = ', num2str(a(3)), ', b = ',
              num2str(b(3))));
87 subplot(3, 3, 9);
88 pie_4 = pie([S(4)/totalPopulation I(4)/totalPopulation R
              (4)/totalPopulation], {'Susceptible', 'Infected', '
              Recovered'});
89 title(strcat('City 4, a = ', num2str(a(4)), ', b = ',
              num2str(b(4))));
90 subplot(3, 3, 5);
91 pie_5 = pie([(S(4)/totalPopulation + S(3)/totalPopulation
              + S(2)/totalPopulation + S(1)/totalPopulation) (I(4)/
              totalPopulation + I(3)/totalPopulation + I(2)/
              totalPopulation + I(1)/totalPopulation) (R(4)/
              totalPopulation + R(3)/totalPopulation + R(2)/
              totalPopulation + R(1)/totalPopulation)], {'
              Susceptible', 'Infected', 'Recovered'});
92 title('Global');
93 drawnow;
94 hold on;
95
96 for clock = 1:clock_max
97     t = clock * dt;
98     % Allow each system to evolve before considering
          changes in population
99     % due to traffic.
100    if (t >= (time_simulated / 8))
101        startedTravel = true;
102    end
103
104    if startedTravel
105        for c = 1:numCities
106            %Consider each susceptible, infected, and
                  recovered individual
107            %Probabilistically move from S to I or from I

```

```

108         to R
109
110         newlyInfected = 0;
111         for s = 1:S(c)
112             if (rand < (dt * a(c) * I(c) / N(c)))
113                 newlyInfected = newlyInfected + 1;
114             end
115         end
116         newlyRecovered = 0;
117         for i = 1:I(c)
118             if (rand < dt * b(c))
119                 newlyRecovered = newlyRecovered + 1;
120             end
121         end
122
123         S(c) = S(c) - newlyInfected;
124         I(c) = I(c) + newlyInfected - newlyRecovered;
125         R(c) = R(c) + newlyRecovered;
126
127     end
128
129     for i = 1:numCities
130         for j = i+1:numCities
131             % Count traffic entering and leaving city
132             % ordered tuple (i, j)
133             if(i ~= j)
134                 % i -> j
135                 initS = S(i);
136                 for s = 1:S(i)
137                     if rand < (TravelSR(i, j) * dt)
138                         && (S(i) ~= 0 && (sum(N) >= R(
139                             j) + I(j) + S(j)))
140                         %Only move from i to j if
141                         % bounds allow one person to
142                         % be removed from i and one
143                         % person
144                         %to be added to j
145                         S(i) = S(i) - 1;
146                         S(j) = S(j) + 1;
147                     end
148                 end
149
150                 for inf = 1:I(i)
151                     if rand < (TravelI(i, j) * dt) &&
152                         (I(i) ~= 0 && (sum(N) >= R(j)
153                             + I(j) + S(j)))
154                         I(i) = I(i) - 1;
155                         I(j) = I(j) + 1;
156                     end
157                 end
158             end
159         end
160     end

```

```

149         end
150
151     for r = 1:R(i)
152         if rand < (TravelSR(i, j) * dt)
153             && (R(i) ~= 0 && (sum(N) >= R
154                 (j) + I(j) + S(j)))
155                 R(i) = R(i) - 1;
156                 R(j) = R(j) + 1;
157         end
158     end
159
160     % j -> i
161
162     for s = 1:S(j)
163         if rand < (TravelSR(j, i) * dt)
164             && (S(j) ~= 0 && (sum(N) >= R(
165                 i) + I(i) + S(i)))
166                 S(j) = S(j) - 1;
167                 S(i) = S(i) + 1;
168         end
169     end
170
171     newS = S(i);
172     for inf = 1:I(j)
173         if rand < (TravelI(j, i) * dt) &&
174             (I(j) ~= 0 && (sum(N) >= R(i)
175                 + I(i) + S(i)))
176                 I(j) = I(j) - 1;
177                 I(i) = I(i) + 1;
178         end
179     end
180
181     for r = 1:R(j)
182         if rand < (TravelSR(j, i) * dt)
183             && (R(j) ~= 0 && (sum(N) >= R
184                 (i) + I(i) + S(i)))
185                 R(j) = R(j) - 1;
186                 R(i) = R(i) + 1;
187         end
188     end
189
190     end
191
192     end
193
194     end
195
196     for i = 1:numCities
197         N_save(i, clock) = S(i)+I(i)+R(i);
198         S_save(i, clock) = S(i);
199         I_save(i, clock) = I(i);
200         R_save(i, clock) = R(i);

```



```

191         I_peaks(1, clock) = I_save(i, clock) + I_peaks
           (1, clock);
192     end
193
194     %Draw pie chart
195     clf('reset')
196
197     subplot(3, 3, 1);
198     pie_1 = pie([S(1)/totalPopulation I(1)/
                  totalPopulation R(1)/totalPopulation], {'
                  Susceptible', 'Infected', 'Recovered'});
199     title(strcat('City 1, a = ', num2str(a(1)), ', b = ',
                  num2str(b(1))));
200     subplot(3, 3, 3);
201     pie_2 = pie([S(2)/totalPopulation I(2)/
                  totalPopulation R(2)/totalPopulation], {'
                  Susceptible', 'Infected', 'Recovered'});
202     title(strcat('City 2, a = ', num2str(a(2)), ', b = ',
                  num2str(b(2))));
203     subplot(3, 3, 7);
204     pie_3 = pie([S(3)/totalPopulation I(3)/
                  totalPopulation R(3)/totalPopulation], {'
                  Susceptible', 'Infected', 'Recovered'});
205     title(strcat('City 3, a = ', num2str(a(3)), ', b = ',
                  num2str(b(3))));
206     subplot(3, 3, 9);
207     pie_4 = pie([S(4)/totalPopulation I(4)/
                  totalPopulation R(4)/totalPopulation], {'
                  Susceptible', 'Infected', 'Recovered'});
208     title(strcat('City 4, a = ', num2str(a(4)), ', b = ',
                  num2str(b(4))));
209     subplot(3, 3, 5);
210     pie_5 = pie([(S(4)/totalPopulation + S(3)/
                  totalPopulation + S(2)/totalPopulation + S(1)/
                  totalPopulation) (I(4)/totalPopulation + I(3)/
                  totalPopulation + I(2)/totalPopulation + I(1)/
                  totalPopulation) (R(4)/totalPopulation + R(3)/
                  totalPopulation + R(2)/totalPopulation + R(1)/
                  totalPopulation)], {'Susceptible', 'Infected', '
                  Recovered'});
211     title('Global');
212     drawnow;
213     hold off;
214
215     end
216
217     %Output static data
218     figure;
219     U = 1.2 * time_simulated;
220     set(gcf, 'double', 'on');

```

```

221 % —— Susceptible —— %
222 L = 1.1 * max(S_save(:));
223 subplot(4, 4, 1);
224 plot(S_save(1,1:clock))
225 title('Susceptible 1');
226 axis([0 U 0 L]);
227 subplot(4, 4, 2);
228 plot(S_save(2,1:clock))
229 title('Susceptible 2');
230 axis([0 U 0 L]);
231 subplot(4, 4, 3);
232 plot(S_save(3,1:clock))
233 title('Susceptible 3');
234 axis([0 U 0 L]);
235 subplot(4, 4, 4);
236 plot(S_save(4,1:clock))
237 title('Susceptible 4');
238 axis([0 U 0 L]);
239
240 % —— Infected —— %
241 L = 1.1 * max(I_save(:));
242 subplot(4, 4, 5);
243 plot(I_save(1,1:clock))
244 title('Infected 1');
245 axis([0 U 0 L]);
246 subplot(4, 4, 6);
247 plot(I_save(2,1:clock))
248 title('Infected 2');
249 axis([0 U 0 L]);
250 subplot(4, 4, 7);
251 plot(I_save(3,1:clock))
252 title('Infected 3');
253 axis([0 U 0 L]);
254 subplot(4, 4, 8);
255 plot(I_save(4,1:clock))
256 title('Infected 4');
257 axis([0 U 0 L]);
258
259 % —— Recovered —— %
260 L = 1.1 * max(R_save(:));
261 subplot(4, 4, 9);
262 plot(R_save(1,1:clock))
263 title('Recovered 1');
264 axis([0 U 0 L]);
265 subplot(4, 4, 10);
266 plot(R_save(2,1:clock))
267 title('Recovered 2');
268 axis([0 U 0 L]);
269 subplot(4, 4, 11);
270 plot(R_save(3,1:clock))

```

```

271 title('Recovered 3');
272 axis([0 U 0 L]);
273 subplot(4, 4, 12);
274 plot(R_save(4,1:clock))
275 title('Recovered 4');
276 axis([0 U 0 L]);
277
278 % — General population — %
279 L = 1.1 * max(N_save(:));
280 subplot(4, 4, 13);
281 plot(N_save(1,1:clock))
282 title('City Pop 1');
283 axis([0 U 0 L]);
284 subplot(4, 4, 14);
285 plot(N_save(2,1:clock))
286 title('City Pop 2');
287 axis([0 U 0 L]);
288 subplot(4, 4, 15);
289 plot(N_save(3,1:clock))
290 title('City Pop 3');
291 axis([0 U 0 L]);
292 subplot(4, 4, 16);
293 plot(N_save(4,1:clock))
294 title('City Pop 4');
295 axis([0 U 0 L]);

```

7 Appendix B: thresholds.m

```
1 clear all;
2 clf;
3 close all;
4
5 numCities = 4;
6 time_simulated = 365 * 2; %number of days
7 clock_max = 365 * 4; %divide number of days into half-day
   intervals
8 dt = time_simulated / clock_max;
9
10 N_save = zeros(numCities, clock_max);
11 S_save = zeros(numCities, clock_max);
12 I_save = zeros(numCities, clock_max);
13 R_save = zeros(numCities, clock_max);
14
15 I_peaks = zeros(1, clock_max);
16
17 N = [1000 500 400 1200];
18 S = [999 498 399 1199];
19 I = [1 2 1 1];
20 R = [0 0 0 0];
21
22 totalPopulation = sum(N);
23
24 %Testing thresholds, R_0 = 1.
25 a = [0.07 0.03 0.06 0.04];
26 b = [0.07 0.03 0.06 0.04];
27 TravelSR = [0 0.1 0.22 0.09; 0.19 0 0.10 0.10; 0.21 0.15
   0 0.20; 0.1 0.2 0.03 0];
28 TravelI = [0 0.05 0.1 0.12; 0.01 0 0.03 0.09; 0.11 0.04 0
   0.09; 0.11 0.10 0.07 0];
29
30 %Testing R_0 > 1, e = 1.
31 %TravelSR = [0 0.013 0.027 0.03; 0.032 0 0.022 0.016;
   0.015 0.04 0 0.05; 0.067 0.076 0.045 0];
32 %TravelI = [0 0.012 0.018 0.019; 0.01 0 0.03 0.09; 0.08
   0.05 0 0.011; 0.051 0.042 0.017 0];
33
34 k = 50;
35 for i = 1:numCities
36     for j = 1:numCities
37         if (i ~= j)
38             TravelSR(i, j) = TravelSR(i, j) + k * 0.003;
39             TravelI(i, j) = TravelI(i, j) + k * 0.003;
40         end
41     end
42 end
```

```

43
44 startedTravel = false;
45 infectedCounter = 0;
46
47 figure;
48 set(gcf, 'double', 'on');
49 subplot(3, 3, 1);
50 pie_1 = pie([S(1)/totalPopulation I(1)/totalPopulation R
    (1)/totalPopulation], {'Susceptible', 'Infected', '
    Recovered'});
51 title(strcat('City 1, a = ', num2str(a(1)), ', b = ',
    num2str(b(1))));
52 subplot(3, 3, 3);
53 pie_2 = pie([S(2)/totalPopulation I(2)/totalPopulation R
    (2)/totalPopulation], {'Susceptible', 'Infected', '
    Recovered'});
54 title(strcat('City 2, a = ', num2str(a(2)), ', b = ',
    num2str(b(2))));
55 subplot(3, 3, 7);
56 pie_3 = pie([S(3)/totalPopulation I(3)/totalPopulation R
    (3)/totalPopulation], {'Susceptible', 'Infected', '
    Recovered'});
57 title(strcat('City 3, a = ', num2str(a(3)), ', b = ',
    num2str(b(3))));
58 subplot(3, 3, 9);
59 pie_4 = pie([S(4)/totalPopulation I(4)/totalPopulation R
    (4)/totalPopulation], {'Susceptible', 'Infected', '
    Recovered'});
60 title(strcat('City 4, a = ', num2str(a(4)), ', b = ',
    num2str(b(4))));
61 subplot(3, 3, 5);
62 pie_5 = pie([(S(4)/totalPopulation + S(3)/totalPopulation
    + S(2)/totalPopulation + S(1)/totalPopulation) (I(4)/
    totalPopulation + I(3)/totalPopulation + I(2)/
    totalPopulation + I(1)/totalPopulation) (R(4)/
    totalPopulation + R(3)/totalPopulation + R(2)/
    totalPopulation + R(1)/totalPopulation)], {'
    Susceptible', 'Infected', 'Recovered'});
63 title('Global');
64 drawnow;
65 hold on;
66
67 for clock = 1:clock_max
68     t = clock * dt;
69     if (t >= (time_simulated / 8))
70         startedTravel = true;
71     end
72
73     if startedTravel
74         for c = 1:numCities

```

```

75         %Consider each susceptible , infected , and
              recovered individual
76         %Probabilistically move from S to I or from I
              to R
77         newlyInfected = 0;
78
79         for s = 1:S(c)
80             if (rand < (dt * a(c) * I(c) / N(c)))
81                 newlyInfected = newlyInfected + 1;
82                 counter = counter + 1;
83             end
84         end
85         newlyRecovered = 0;
86         for i = 1:I(c)
87             if (rand < dt * b(c))
88                 newlyRecovered = newlyRecovered + 1;
89             end
90         end
91
92         S(c) = S(c) - newlyInfected;
93         I(c) = I(c) + newlyInfected - newlyRecovered;
94         R(c) = R(c) + newlyRecovered;
95
96     end
97 end
98
99 for i = 1:numCities
100     for j = i+1:numCities
101         % Count traffic entering and leaving
              ordered tuple (i, j)
102         if(i ~= j)
103             % i -> j
104             initS = S(i);
105             for s = 1:S(i)
106                 if rand < (TravelSR(i, j) * dt)
107                     && (S(i) ~= 0 && (sum(N) >= R(
                          j) + I(j) + S(j)))
108                     %Only move from i to j if
                          bounds allow one person to
                          be removed from i and one
                          person
109                     %to be added to j
110                     S(i) = S(i) - 1;
111                     S(j) = S(j) + 1;
112                 end
113             end
114         end
115         for inf = 1:I(i)
116             if rand < (TravelI(i, j) * dt) &&
                  (I(i) ~= 0 && (sum(N) >= R(j)

```

```

116         + I(j) + S(j))
117         I(i) = I(i) - 1;
118         I(j) = I(j) + 1;
119     end
120 end
121 for r = 1:R(i)
122     if rand < (TravelSR(i, j) * dt)
123         && (R(i) ~= 0 && (sum(N) >= R(
124             j) + I(j) + S(j)))
125         R(i) = R(i) - 1;
126         R(j) = R(j) + 1;
127     end
128 end
129 % j -> i
130 for s = 1:S(j)
131     if rand < (TravelSR(j, i) * dt)
132         && (S(j) ~= 0 && (sum(N) >= R(
133             i) + I(i) + S(i)))
134         S(j) = S(j) - 1;
135         S(i) = S(i) + 1;
136     end
137 end
138 newS = S(i);
139 for inf = 1:I(j)
140     if rand < (TravelI(j, i) * dt) &&
141         (I(j) ~= 0 && (sum(N) >= R(i)
142             + I(i) + S(i)))
143         I(j) = I(j) - 1;
144         I(i) = I(i) + 1;
145     end
146 end
147 for r = 1:R(j)
148     if rand < (TravelSR(j, i) * dt)
149         && (R(j) ~= 0 && (sum(N) >= R(
150             i) + I(i) + S(i)))
151         R(j) = R(j) - 1;
152         R(i) = R(i) + 1;
153     end
154 end
155 end
156 end

```

```

157     for i = 1:numCities
158         N_save(i, clock) = S(i)+I(i)+R(i);
159         S_save(i, clock) = S(i);
160         I_save(i, clock) = I(i);
161         R_save(i, clock) = R(i);
162         I_peaks(1, clock) = I_save(i, clock) + I_peaks
            (1, clock);
163     end
164
165     %Draw pie charts
166     clf('reset')
167
168     subplot(3, 3, 1);
169     pie_1 = pie([S(1)/totalPopulation I(1)/
        totalPopulation R(1)/totalPopulation], {'
        Susceptible', 'Infected', 'Recovered'});
170     title(strcat('City 1, a = ', num2str(a(1)), ', b = ',
        num2str(b(1))));
171     subplot(3, 3, 3);
172     pie_2 = pie([S(2)/totalPopulation I(2)/
        totalPopulation R(2)/totalPopulation], {'
        Susceptible', 'Infected', 'Recovered'});
173     title(strcat('City 2, a = ', num2str(a(2)), ', b = ',
        num2str(b(2))));
174     subplot(3, 3, 7);
175     pie_3 = pie([S(3)/totalPopulation I(3)/
        totalPopulation R(3)/totalPopulation], {'
        Susceptible', 'Infected', 'Recovered'});
176     title(strcat('City 3, a = ', num2str(a(3)), ', b = ',
        num2str(b(3))));
177     subplot(3, 3, 9);
178     pie_4 = pie([S(4)/totalPopulation I(4)/
        totalPopulation R(4)/totalPopulation], {'
        Susceptible', 'Infected', 'Recovered'});
179     title(strcat('City 4, a = ', num2str(a(4)), ', b = ',
        num2str(b(4))));
180     subplot(3, 3, 5);
181     pie_5 = pie([(S(4)/totalPopulation + S(3)/
        totalPopulation + S(2)/totalPopulation + S(1)/
        totalPopulation) (I(4)/totalPopulation + I(3)/
        totalPopulation + I(2)/totalPopulation + I(1)/
        totalPopulation) (R(4)/totalPopulation + R(3)/
        totalPopulation + R(2)/totalPopulation + R(1)/
        totalPopulation)], {'Susceptible', 'Infected', '
        Recovered'});
182     title('Global');
183     drawnow;
184     hold off;
185
186 end

```



```

187
188 %Output static data
189 figure;
190 title(strcat(strcat(strcat('dt = ', num2str(dt)), ' k = '),
    , num2str(k)));
191 set(gcf, 'double', 'on');
192 U = 1.2 * time_simulated;
193 x = 1:clock;
194 L = 1.1 * max(I_save(:), max(S_save(:), R_save(:)));
195
196 subplot(2, 4, 1);
197 plot(x, S_save(1, 1:clock), x, I_save(1, 1:clock), x,
    R_save(1, 1:clock));
198 title('City 1');
199 xlabel('Time');
200 ylabel('Num. of people');
201
202 subplot(2, 4, 2);
203 plot(x, S_save(2, 1:clock), x, I_save(2, 1:clock), x,
    R_save(2, 1:clock));
204 title('City 2');
205 xlabel('Time');
206 ylabel('Num. of people');
207 subplot(2, 4, 3);
208 plot(x, S_save(3, 1:clock), x, I_save(3, 1:clock), x,
    R_save(3, 1:clock));
209 title('City 3');
210 xlabel('Time');
211 ylabel('Num. of people');
212 subplot(2, 4, 4);
213 plot(x, S_save(4, 1:clock), x, I_save(4, 1:clock), x,
    R_save(4, 1:clock));
214 title('City 4');
215 xlabel('Time');
216 ylabel('Num. of people');
217
218 % — General population — %
219 L = 1.1 * max(N_save(:));
220 subplot(2, 4, 5);
221 plot(N_save(1, 1:clock))
222 title('Total Population in City 1');
223 xlabel('Time');
224 ylabel('Num. of people');
225 axis([0 U 0 L]);
226 subplot(2, 4, 6);
227 plot(N_save(2, 1:clock))
228 title('Total Population in City 2');
229 xlabel('Time');
230 ylabel('Num. of people');
231 axis([0 U 0 L]);

```

```

232 subplot(2, 4, 7);
233 plot(N_save(3,1:clock))
234 title('Total Population in City 3');
235 xlabel('Time');
236 ylabel('Num. of people');
237 axis([0 U 0 L]);
238 subplot(2, 4, 8);
239 plot(N_save(4,1:clock))
240 title('Total Population in City 4');
241 xlabel('Time');
242 ylabel('Num. of people');
243 axis([0 U 0 L]);

```

8 Appendix C: dynamicRates.m

```

1
2 clear all;
3 clf;
4 close all;
5
6 numCities = 4;
7 time_simulated = 365; %number of days
8 clock_max = 365; %divide number of days into day
   intervals
9 dt = time_simulated / clock_max;
10
11 N_save = zeros(numCities, clock_max);
12 S_save = zeros(numCities, clock_max);
13 I_save = zeros(numCities, clock_max);
14 R_save = zeros(numCities, clock_max);
15 I_peaks = zeros(1, clock_max);
16
17 N = [1000 500 400 1200];
18 S = [999 498 399 1199];
19 I = [1 2 1 1];
20 R = [0 0 0 0];
21
22 totalPopulation = sum(N);
23
24 a = [0.15 0.12 0.09 0.11]; % infectivity a = # of new
   cases per day caused by one infected person.
25 b = [0.01 0.01 0.01 0.01]; %time taken to recover per
   person is 1/b.
26
27 TravelSR = [0 0.1 0.3 0.09; 0.19 0 0.10 0.10; 0.29 0.15 0
   0.20; 0.1 0.2 0.03 0];
28 TravelI = [0 0.05 0.1 0.12; 0.01 0 0.03 0.09; 0.11 0.04 0
   0.09; 0.11 0.10 0.07 0];
29
30 startedTravel = false;

```

```

31
32 figure;
33 set(gcf, 'double', 'on');
34 subplot(3, 3, 1);
35 pie_1 = pie([S(1)/totalPopulation I(1)/totalPopulation R
              (1)/totalPopulation], {'Susceptible', 'Infected', '
              Recovered'});
36 title(strcat('City 1, a = ', num2str(a(1)), ', b = ',
              num2str(b(1))));
37 subplot(3, 3, 3);
38 pie_2 = pie([S(2)/totalPopulation I(2)/totalPopulation R
              (2)/totalPopulation], {'Susceptible', 'Infected', '
              Recovered'});
39 title(strcat('City 2, a = ', num2str(a(2)), ', b = ',
              num2str(b(2))));
40 subplot(3, 3, 7);
41 pie_3 = pie([S(3)/totalPopulation I(3)/totalPopulation R
              (3)/totalPopulation], {'Susceptible', 'Infected', '
              Recovered'});
42 title(strcat('City 3, a = ', num2str(a(3)), ', b = ',
              num2str(b(3))));
43 subplot(3, 3, 9);
44 pie_4 = pie([S(4)/totalPopulation I(4)/totalPopulation R
              (4)/totalPopulation], {'Susceptible', 'Infected', '
              Recovered'});
45 title(strcat('City 4, a = ', num2str(a(4)), ', b = ',
              num2str(b(4))));
46 subplot(3, 3, 5);
47 pie_5 = pie([(S(4)/totalPopulation + S(3)/totalPopulation
              + S(2)/totalPopulation + S(1)/totalPopulation) (I(4)/
              totalPopulation + I(3)/totalPopulation + I(2)/
              totalPopulation + I(1)/totalPopulation) (R(4)/
              totalPopulation + R(3)/totalPopulation + R(2)/
              totalPopulation + R(1)/totalPopulation)], {'
              Susceptible', 'Infected', 'Recovered'});
48 title('Global');
49 hold on;
50
51 for clock = 1:clock_max
52     t = clock * dt;
53     % Allow each system to evolve before considering
54     % changes in population
55     % due to traffic.
56     if (t >= (time_simulated / 8))
57         startedTravel = true;
58     end
59     if startedTravel
60         for c = 1:numCities
61             %Consider each susceptible, infected, and

```

```

62         recovered individual
        %Probabilistically move from S to I or from I
           to R
63
64         newlyInfected = 0;
65
66         for s = 1:S(c)
67             if (rand < (dt * a(c) * I(c) / N(c)))
68                 dt * a(c) * I(c) / N(c)
69                 newlyInfected = newlyInfected + 1;
70             end
71         end
72         newlyRecovered = 0;
73         for i = 1:I(c)
74             if (rand < dt * b(c))
75                 dt * b(c)
76                 newlyRecovered = newlyRecovered + 1;
77             end
78         end
79
80         S(c) = S(c) - newlyInfected;
81         I(c) = I(c) + newlyInfected - newlyRecovered;
82         R(c) = R(c) + newlyRecovered;
83     end
84 end
85
86 for i = 1:numCities
87     for j = i+1:numCities
88         % Count traffic entering and leaving
           ordered tuple (i, j)
89         if(i ~= j)
90
91             biasedSR_j = 0;
92             biasedI_j = 0;
93             biasedSR_i = 0;
94             biasedI_i = 0;
95             if (S(i) + R(i)) < I(i)
96                 %Bias rates
97                 biasedSR_j = (I(i) / N(i));
98                 biasedI_j = (I(i) / N(i));
99
100                 biasedSR_i = 1 - biasedSR_j;
101                 biasedI_i = 1 - biasedI_j;
102
103             else
104                 %Resume static rates
105                 biasedSR_j = TravelSR(i, j);
106                 biastedI_j = TravelI(i, j);
107
108                 biasedSR_i = TravelSR(j, i);

```

```

109         biasedI_i = TravelI(j, i);
110
111     end
112
113     % i -> j
114     for s = 1:S(i)
115         if rand < biasedSR_j && (S(i) ~=
            0) && (sum(N) >= R(j) + I(j) +
            S(j))
116             %Only move from i to j if
            bounds allow one person to
            be removed from i and one
            person
            %to be added to j
            S(i) = S(i) - 1;
            S(j) = S(j) + 1;
117         end
118     end
119
120     for inf = 1:I(i)
121         if rand < biasedI_j && (I(i) ~=
            0) && (sum(N) >= R(j) + I(j) +
            S(j))
122             I(i) = I(i) - 1;
123             I(j) = I(j) + 1;
124         end
125     end
126
127     for r = 1:R(i)
128         if rand < biasedSR_j && (R(i) ~=
            0) && (sum(N) >= R(j) + I(j)
            + S(j))
129             R(i) = R(i) - 1;
130             R(j) = R(j) + 1;
131         end
132     end
133
134     % j -> i
135     for s = 1:S(j)
136         if rand < biasedSR_i && (S(j) ~=
            0) && (sum(N) >= R(i) + I(i) +
            S(i))
137             S(j) = S(j) - 1;
138             S(i) = S(i) + 1;
139         end
140     end
141
142     for inf = 1:I(j)
143         if rand < biasedI_i && (I(j) ~=

```

```

                                0) && (sum(N) >= R(i) + I(i) +
                                S(i))
148                                I(j) = I(j) - 1;
149                                I(i) = I(i) + 1;
150                                end
151                                end
152
153                                for r = 1:R(j)
154                                    if rand < biasedSR_i && (R(j) ~=
                                        0) && (sum(N) >= R(i) + I(i)
                                        + S(i))
155                                        R(j) = R(j) - 1;
156                                        R(i) = R(i) + 1;
157                                    end
158                                end
159                                end
160                                end
161                                end
162                                end
163
164                                for i = 1:numCities
165                                    N_save(i, clock) = S(i)+I(i)+R(i);
166                                    S_save(i, clock) = S(i);
167                                    I_save(i, clock) = I(i);
168                                    R_save(i, clock) = R(i);
169                                    I_peaks(1, clock) = I_save(i, clock) + I_peaks
                                        (1, clock);
170                                end
171
172                                %Draw pie charts
173                                clf('reset')
174
175                                subplot(3, 3, 1);
176                                pie_1 = pie([S(1)/totalPopulation I(1)/
                                    totalPopulation R(1)/totalPopulation], {'
                                    Susceptible', 'Infected', 'Recovered'});
177                                title(strcat('City 1, a = ', num2str(a(1)), ', b = ',
                                    num2str(b(1))));
178                                subplot(3, 3, 3);
179                                pie_2 = pie([S(2)/totalPopulation I(2)/
                                    totalPopulation R(2)/totalPopulation], {'
                                    Susceptible', 'Infected', 'Recovered'});
180                                title(strcat('City 2, a = ', num2str(a(2)), ', b = ',
                                    num2str(b(2))));
181                                subplot(3, 3, 7);
182                                pie_3 = pie([S(3)/totalPopulation I(3)/
                                    totalPopulation R(3)/totalPopulation], {'
                                    Susceptible', 'Infected', 'Recovered'});
183                                title(strcat('City 3, a = ', num2str(a(3)), ', b = ',
                                    num2str(b(3))));

```

```

184     subplot(3, 3, 9);
185     pie_4 = pie([S(4)/totalPopulation I(4)/
                  totalPopulation R(4)/totalPopulation], {'
                  Susceptible', 'Infected', 'Recovered'});
186     title(strcat('City 4, a = ', num2str(a(4)), ', b = ',
                  num2str(b(4))));
187     subplot(3, 3, 5);
188     pie_5 = pie([(S(4)/totalPopulation + S(3)/
                  totalPopulation + S(2)/totalPopulation + S(1)/
                  totalPopulation) (I(4)/totalPopulation + I(3)/
                  totalPopulation + I(2)/totalPopulation + I(1)/
                  totalPopulation) (R(4)/totalPopulation + R(3)/
                  totalPopulation + R(2)/totalPopulation + R(1)/
                  totalPopulation)], {'Susceptible', 'Infected', '
                  Recovered'});
189     title('Global');
190     drawnow;
191     hold off;
192
193 end
194
195 %Output static data
196 figure;
197 subplot(4, 4, 1);
198 plot(S_save(1,1:clock))
199 title('Susceptible 1');
200 subplot(4, 4, 2);
201 plot(S_save(2,1:clock))
202 title('Susceptible 2');
203 subplot(4, 4, 3);
204 plot(S_save(3,1:clock))
205 title('Susceptible 3');
206 subplot(4, 4, 4);
207 plot(S_save(4,1:clock))
208 title('Susceptible 4');
209 subplot(4, 4, 5);
210 plot(I_save(1,1:clock))
211 title('Infected 1');
212 subplot(4, 4, 6);
213 plot(I_save(2,1:clock))
214 title('Infected 2');
215 subplot(4, 4, 7);
216 plot(I_save(3,1:clock))
217 title('Infected 3');
218 subplot(4, 4, 8);
219 plot(I_save(4,1:clock))
220 title('Infected 4');
221 subplot(4, 4, 9);
222 plot(R_save(1,1:clock))
223 title('Recovered 1');

```

```

224 subplot(4, 4, 10);
225 plot(R_save(2,1:clock))
226 title('Recovered 2');
227 subplot(4, 4, 11);
228 plot(R_save(3,1:clock))
229 title('Recovered 3');
230 subplot(4, 4, 12);
231 plot(R_save(4,1:clock))
232 title('Recovered 4');
233 subplot(4, 4, 13);
234 plot(N_save(1,1:clock))
235 title('City Pop 1');
236 subplot(4, 4, 14);
237 plot(N_save(2,1:clock))
238 title('City Pop 2');
239 subplot(4, 4, 15);
240 plot(N_save(3,1:clock))
241 title('City Pop 3');
242 subplot(4, 4, 16);
243 plot(N_save(4,1:clock))
244 title('City Pop 4');

```