DEVELOPMENT, IMPLEMENTATION, AND ANALYSIS OF A CONTACT MODEL FOR AN INFECTIOUS DISEASE

Brett Morinaga Thompson, B.A.

Thesis Prepared for the Degree of

MASTER OF SCIENCE

UNIVERSITY OF NORTH TEXAS

May 2009

APPROVED:

Armin Mikler, Major Professor
Tom Jacob, Committee Member
Paul Tarau, Committee Member
Krishna Kavi, Chair of the Department of Computer Science and Engineering
Costas Tsatsoulis, Dean of College of Engineering
Michael Monticino, Interim Dean of the Robert B. Toulouse School of Graduate Studies

Thompson, Brett Morinaga. Development, Implementation, and Analysis of a Contact Model for an Infectious Disease. Master of Science (Computer Science), May 2009, 47 pp., 17 tables, 20 figures, bibliography, 13 titles.

With a growing concern of an infectious diseases spreading in a population, epidemiology is becoming more important for the future of public health. In the past epidemiologist used existing data of an outbreak to help them determine how an infectious disease might spread in the future. Now with computational models, they able to analysis data produced by these models to help with prevention and intervention plans. This paper looks at the design, implementation, and analysis of a computational model based on the interactions of the population between individuals. The design of the working contact model looks closely at the SEIR model used as the foundation and the two timelines of a disease. The implementation of the contact model is reviewed while looking closely at data structures. The analysis of the experiments provide evidence this contact model can be used to help epidemiologist study the spread of an infectious disease based on the contact rate of individuals.

ACKNOWLEDGMENTS

I take great pleasure to thank those who made this thesis possible.

I want to express my gratitude to major professor, Dr. Armin Mikler. With his ideas, patience, guidance, clear explanations of the research, he helped me to put his ideas to work and create a research that may used for years to come.

This paper would not have been possible without Bosko's hard work in developing the original code of the contact model. He's expertise in programming the contact model provided a starting point for this paper.

I am grateful for the dedication the graduate students of the Computational Epidemiology Research Lab (CERL) at the University of North Texas. Specifically I would like to thank Angel Bravo-Salgado's hard work in helping me test features of the contact model and analysis the results. I would also like to thank Aliasgar Amin for his determination in implementing new features and developing new ideas on to accomplish this feat.

I wish to thank my family and friends for their support during these long months for completion in my work. My sister and my parents were particular supportive and my loving friend and teacher Silver Ra for his gracious, motivating support.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1. Discussion of the Problem

In today's modern world where a person can easily travel halfway around the world in a matter hours, the ease in which an infectious disease may spread in the world population has created many concerns in the public health sector. With this growing concern of an infectious diseases spreading in a population, epidemiology is becoming more important for the future of public health. Preparedness for epidemics and pandemics has become a high priority in many countries [9]. In the past, experts like epidemiologist have used existing data from previous outbreaks and epidemics to study how the infectious disease spread through the population. The irony of this is if experts are to learn about how and why an infectious disease spreads an outbreak must occur. This is no longer feasible in the case of newly emerging infectious diseases or a bio-terror attack. If we are to fully understand and analyze how infectious diseases spread through a population before a sudden outbreak or bio-terror attack, we need to have the proper analysis tools for the experts in the fields of public health and epidemiology to study.

Giving the complexity of this task, mathematical and computational models are needed to create and design the right tools so that predictions and comparisons of the effects of different preparedness strategies can be assessed [12]. This is where computer science and computer scientist play a major role in helping to address this problem. Using the help of epidemiologists, computer scientist are able to create computational and mathematical models to simulate the spread of an infectious disease based on specific characteristics of an infectious disease. The resulting data is analyzed to determine whether or not an epidemic could occur with the specific characteristics. Epidemiologists and public health experts then

use the statistical data produced by these models to make decisions in planning, prevention, and surveillance of an infectious disease to ensure an epidemic's impact to the world population is minimal.

1.1.1. Epidemiology

Epidemiology is the study of the factors of a disease affecting the health and illness of populations. These factors include the causes, distribution, and control of a particular disease [11]. The sum of these factors show the presence or the absence of that disease. Epidemiology serves as the foundation of the logical interventions made in the interest of the public health and preventive medicine. Experts use epidemiology for identifying risk factors of diseases and determining the optimal treatment approaches necessary for clinical practice [6]. Epidemiology also includes the development of statistical correlations to indicate the degree of risk that someone with a particular exposure pattern or lifestyle has of contracting a specific disease.

Epidemiology in the past has relied on existing data from previous outbreaks to base decisions in the area of planning, prevention, and surveillance. With the emerging of new diseases and viruses, the existing data may have become out-dated and useless. In computational epidemiology, computer scientist and epidemiologist are developing new tools to apply past knowledge of outbreaks with the computing power of computers. The combination of these new tools and techniques create a better understanding of how and why infectious diseases spread in a population. Using this knowledge, epidemiologist can now work with public health officials to design more effective planning, prevention, and surveillance programs world-wide.

1.1.2. Contact Model

A contact model is a statistical model of a population in which each individual person of the population interacts with a set number of individuals during a specific time period. Using different input parameters to match a specific infectious disease, an epidemiologist can get a better understanding of how an infectious disease might spread based on the contact rate

of the individuals in the population. In order to model the spread of an infectious disease in a population with high fidelity, several key characteristics that are relevant to an outbreak of a disease must be taken into account. There are also characteristics of the population that need to be taken into account. There are population characteristics and characteristics of each individual person. The disease and population characteristics are described later in Chapter 2.

1.2. Computational Model vs. Mathematical Model

A computer program that can be run on one or more machines that simulates a theoretical model of a specific system is a computational model. A theoretical model that uses mathematical language or logic to explain a specific system is a mathematical model. A mathematical model uses underlying assumptions while a computational model uses known facts as a foundation of the model.

The potential impact of control and interventions procedures for an infectious disease along with a population's dynamics of the transmission of an infectious disease are essential to understand in mathematical models [10]. A tremendous amount extrapolation is used to develop mathematical equations to create a working model. This results in a complexity that relies on the underlying assumptions used in developing the model.

Given specific input parameters, computational models are used to gain a better understanding of multiple processes seen in the natural world [5]. Simulated simultaneously, these set of processes can be used to predict the outcome of the original natural process. The more processes in the set being simulated, the more difficult is for others to understand and analyze the results because of the ambiguity of the data [10].

1.3. Existing Approaches

There have been numerous mathematical models to date that simulate an infectious disease. There are some that simulate a specific disease such as influenza or HIV. However, few mathematical model exist that simulate a general epidemic or pandemic.

One of these [1], uses the susceptible-infectious-removed (SIR) model approach. This approach assumes a susceptible individual becomes infectious when exposed, leaving out the critical symptomatic period. Figure 1.1 shows a state diagram of the SIR model. In [1], the authors use a stochastic SIR approach that creates contacts between susceptible and infectious individuals based on local and global contacts according to a contact distribution centered on the infective individual. This model determines the threshold behavior and final outcome of stochastic SIR epidemics with in a global epidemic.

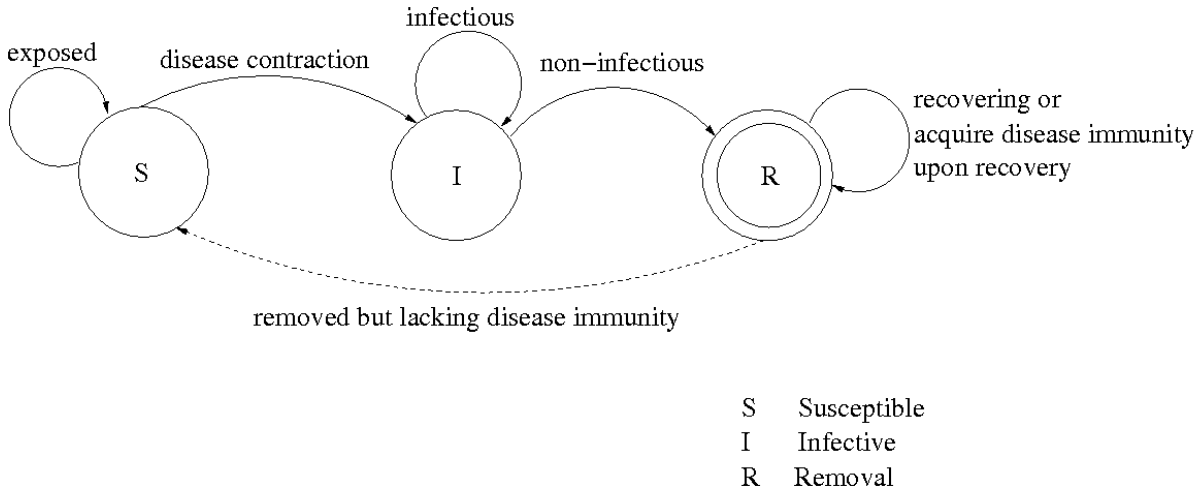

FIGURE 1.1. SIR State Diagram

Most models that simulate an infectious disease assume the incubation period of a disease is negligible. This means when a susceptible individuals become infected they instantaneously are infectious and later recover. The SIR model is the classic approach to creating this. These compartmental models as known as SIR or SIRS model have been widely studied in the past, [1], [4], [2]. The authors in [8] have used a new approach in which the symptomatic period is considered. The method uses the susceptible-latent-infective-asymptomatic-recovered (SLIAR) model based on Brauer's approach [3]. Here the authors show that the predictions of stochastic simulations of network models can be acquied through simple deterministic compartmental models. By showing this, new predictions were formulated to show that the require number of doses of antiviral treatment is sensitive to the initial number of

infected individuals. This can be used for pre-epdemics vaccination and for treatment during an epidemic.

CHAPTER 2

OVERVIEW OF THE CONTACT MODEL

2.1. Structure of the Contact Model

Today the main reason why an infectious disease spreads through a population quickly is because of individuals' interactions with different people on a daily basis. The majority of the world population live in densely populated areas. The amount of contacts people have on a daily basis is much higher than it was in the past. The time for an infectious disease to spread through a dense population in a metroplex is the fastest it has ever been. Thus it is important to follow the contact behavior of infectious individuals for analysis of an outbreak in a fixed population.

The contact model simulates an infectious individual's contact behavior in a fixed population size. Individuals' contact behaviors are modeled with the contact rate (CR). The contact rate is the number of times an infectious individual comes into contact with a susceptible individual. By recording each infectious individual's contact behavior in a fixed susceptible population, the model can simulate the spread of a disease in the population based on an individual's contact rate.

The majority of epidemiological systems use the susceptible-infectious-removal (SIR) class of models. The population is categorized as either susceptible, infectious, or recovered. Expansion of these systems have been refined to include an additional class of exposed individuals. These individuals are exposed but not yet infectious. The new classes are susceptible-exposed-infectious-recovered (SEIR) creating two different timelines for a disease.

The structure of the contact model uses the SEIR model approach to take into account the the two timelines of a disease. These are the infectious and symptomatic timelines as shown in Figure 2.1. Table 2.1 list the characteristics of a disease that are associated with

the life cycle of a disease. These characteristics are important to consider when modeling the life cycle or timelines of a disease. Figure 2.2 shows the state diagram for the SEIR model.

The first timeline, the infectious timeline, follows the effect of a disease in an individual.

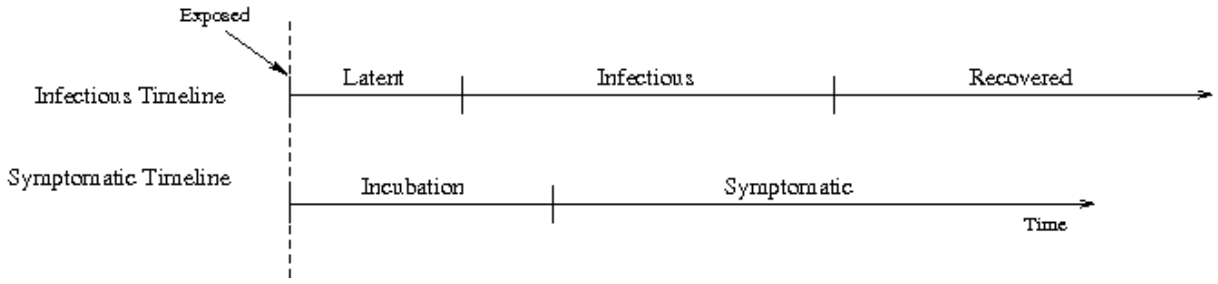

FIGURE 2.1. Disease Timelines

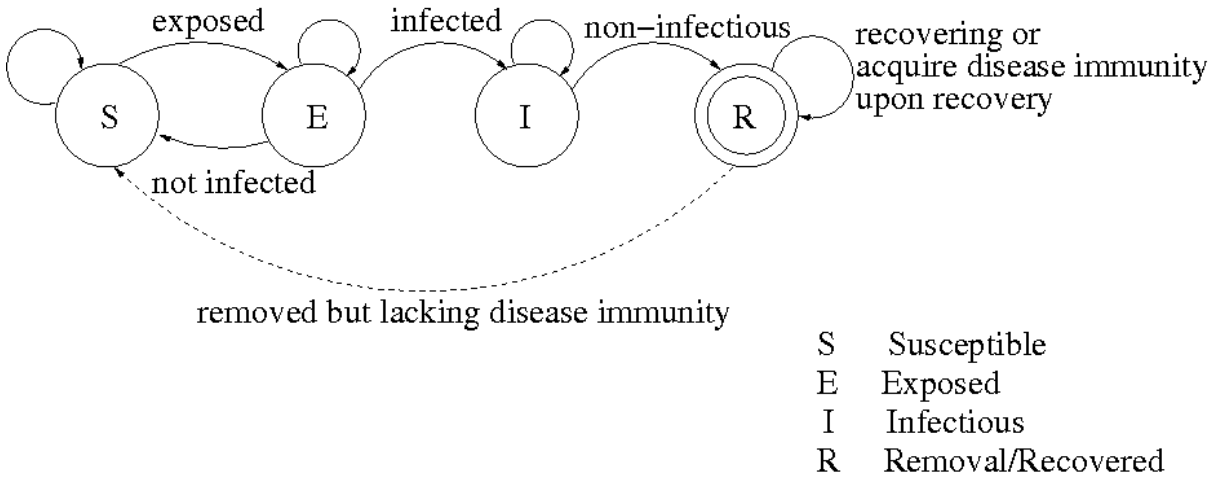

FIGURE 2.2. SEIR State Diagram

In this timeline, a person starts off being susceptible. Once exposed to the disease, the individual becomes infected starting the latent period of the disease. After the latent period is over, they become infectious. In the infectious period, the model keeps track of how many people the infected individual comes into contact with. The infected person's contact rate is entered as an input value before the start of the simulation. When the infectious period is completed the individual moves into the non-infectious or the recovered state, ending the life cycle of the disease in that person.

TABLE 2.1. Disease Input Parameters

Parameter Name	Description
latent period	The time between when a person becomes infected to when they become infectious.
incubation period	The time between when a person becomes infected to the onset of symptoms.
infectious period	The time period when a person is infectious.
infectivity	The probability of a disease or virus establishing an infection.

The second timeline, the symptomatic timeline, follows the incubation period of the disease. Once the individual becomes infected and starts the latent period of the disease, the incubation period is started as well. The incubation period will determine when the individual becomes symptomatic. At the end of the incubation period, the individual becomes symptomatic. The individual remains symptomatic until they enter the non-infectious or recovered state. It is important to consider the contact behavior of an individual who is displaying symptoms. When a person becomes sick or shows signs of being sick, they tend to self-quarantine themselves from other people. So as a person exhibits symptoms of a illness, their interaction with people will decrease. The more sick the person becomes, the more their contact behavior will slow. To model this behavior, when an individual becomes symptomatic, the contact rate of that individual decreases by a percentage set by an input value, also known as Δ contact rate (DCR). The individual's contact rate decreases every day until they become non-infectious or recovered. If a disease has multiple symptoms that are exhibited at the beginning of the symptomatic period, this can be modeled by increasing the DCR. A disease's timeline is an important component in determining how it spreads through the population. For example, if the latent period is three days long, the infected

individual will not be able to spread the disease until four days after coming in contact with an infectious individual.

TABLE 2.2. Individual Input Parameters

Parameter Name	Description
contact rate (CR)	Determines how many people each individual comes into contact with.
Δ contact rate (DCR)	The rate at which a person's CR decreases after they are symptomatic.
resistance	A person's resistance level to the disease.
immune level	A person's immunity level based on their immune system

2.2. Computing Timing

The contact model has multiple layers that result in a simulation that takes into account several key components of the spread of an infectious disease. This requires plenty of computing power in order to accomplish this in enough time to analysis the results and to provide intervention and prevention mechanisms to health officials before an outbreak occurs. The starting population size is a key factor in determining the amount of time it takes to execute a simulation. Having a big population creates more possibilities to generate infectious individuals which effects the amount of contacts per cycle. As the infectious population grows, the amount of data being recorded grows as well slowing the overall execution time of a simulation. The input and output of the simulator produces more overhead than the execution of the code itself. It is important to consider the amount of computing available when deciding the population size of a simulation so there can be enough time to analysis results for future simulations.

2.3. Random Numbers

The model only focuses on the spread of an infectious disease based a person's contact with other individuals in the population. Using random numbers to decide what person

gets picked from the population allows for indiscriminate selection of the population, not similar to what an infectious disease does in a real world situation. Since people tend to make contacts with individuals in a network of clusters, the model simplifies this scenario by using random sampling of probability distribution in the random experiments. This is very similar to the class of computational algorithms used in the Monte Carlo method. The simulator has a define set of possible numbers. A pseudo-random number generator creates a uniform distribution of this set of numbers. The simulator then aggregates the result of each random experiment into the final result.

2.4. Storage

During a simulation, the amount of volatile memory the contact model requires is dependent on the size of the population. Each person in the population has a fixed number of variables. A large population will have a large amount of variables to be stored in memory. Therefore the population variables determine the overall size of the memory needed to store the population in volatile storage.

The amount of non-volatile memory the contact model requires for the output data is dependent on the size of the population. Since every variable for every individual is recorded in a file, the output file that records the data of the population grows in size in proportion to the size of the population. The size of the output file that records the data for producing the graphs grows as the overall runtime of a simulation increases. The longer the runtime of a simulation, more data is recorded increasing the size of the output file.

CHAPTER 3

DESIGN AND IMPLEMENTATION

3.1. Code Design

The contact model incorporates specific human behavior and disease characteristics that contribute to the spread of an infectious disease. When studying a past epidemic or outbreak of a disease, epidemiologist take into consideration key components that directly contribute to the outbreak. They have identified key characteristics that some diseases have in common. In the contact model these disease characteristics are known as the disease states. Table 3.1 list the disease states used in the contact model. All of these characteristics are major factors

TABLE 3.1. Disease States

State	Description
susceptible	At risk of becoming infected.
latent	In the latent period.
infectious	In the infectious period.
non_infectious	No longer infectious and susceptible to the disease.
vaccinated	Has received vaccination and cannot become infected.
immune	Cannot become infected.

of how a disease progresses in a human body. In designing the model, all of these components that reflect reality, need to be simplified in a way that decreases complexity of the model but at the same time fulfills the simulator objectives [7].

Epidemiologist have identified specific human behaviors that directly contribute to an outbreak. They include a person's rate of interaction between groups of people before and during the symptomatic period, a person's immunity level, and a person's resistance level.

When a person begins to show signs of being sick, their daily activities are usually decreased. Depending on the severity of the symptoms, a sick person could self-isolated themselves from coming into contact with the least amount of people, reducing the spread the of the disease. A person's immunity level plays an important role in determining how long it takes for a person to become infected after being exposed or how long it takes for that person to recover. A person's resistance level plays a more direct role in determining whether the person becomes infected after exposure. The condition of a person's immune system or the overall health of a person contributes to how resistant that person's body is to an infectious disease. This model assumes these human behaviors at the most basic level to decrease complexity of the model. A person's immunity level and resistance level has been added to the code, but has not been fully implemented and tested for use in experiments so the impact they had on the design process will only be discussed.

3.1.1. Overall Design of the Code

The overall design of the code uses the SEIR deterministic model discussed in section 2.1. This approach was chosen to add an infected but not yet infectious state to take into account the symptomatic stage of being infected by an infectious disease. Infectious diseases require time in the environment or host to develop an infectious stage. This state is known as the latent period. It is important to take into account the latent period of a disease because it affects the rate at which the disease spreads throughout the population. By adding the latent period of a disease, many different infectious diseases can be modeled by the simulator increasing the applications the simulator could be used for by epidemiologist.

Once the infectious stage has developed, the host or infected person is infectious only for a specific period of time while their immune system fights off the disease. This stage is know as the infectious period. In some cases, the infected individual does not recover during or after the infectious period. Specific pathogenic organisms of infectious diseases take time to progress in an infected host before the host starts to exhibit symptoms of the particular

disease. The time elapsed between exposure of the pathogenic organisms and when the host show signs of the clinical onset of the disease is known as the incubation period.

Most infectious diseases have two timelines in its life cycle as discussed in section 2.1. Within each of these timelines the disease migrate through different stages of its life in the human body. For example, the latent, incubation, and infectious periods. In order to model the two timelines of a disease, each person in the population during the simulation has a disease state assigned to them. Disease states are assigned during the initialization of the population or during the simulation on an individual basis as a person moves through the different stages of the disease. This is discussed in detail in section 3.2.1. Each person also has two counters that keep track of how long a person has been in a particular state. Two counters are needed to model both timelines simultaneously. They are called time of disease state (*timeOfDS*) and time of incubation (*timeOfIncubation*). Both counters start when a person becomes latent. The *timeOfDS* is used to model the infectious timeline and is reset when the person becomes infectious in order to monitor how long that person stays in the infectious period. The *timeOfIncubation* is used to model the symptomatic timeline. Once the counter equals the incubation period the person becomes symptomatic.

Another feature that was added allows the user to input a number of how many people in the population are vaccinated. The simulator will randomly assign individuals the *vaccinated* disease state. A person who has a *vaccinated* disease state can not be infected during the entire simulation. Even though vaccinations do not work one hundred percent of the time, for simplicity this model assumes that they do. This feature allows an epidemiologist to simulate a scenario when vaccinations of a specific disease are used in the prevention planning of an epidemic. An epidemiologist can look at how different immune proportions of the population can effect the overall spread of the disease and determine the percentage of the population needed to be vaccinated to reduce the spread of the disease.

To create more variance in how an individual becomes infected, the resistance and the immune level variables were added to the code even though they have not been fully implemented and tested. The simulator adds the same resistance and the immune level to every individual in the population. In the future, these features could be assigned different values for each individual based on certain physiological properties of a person. For instance, a person's age can contribute to how resistant they are to a disease. Another example would be to assign a person's immune level randomly because not every one person's immune system is at the same level of effectiveness.

In general, one of the things to consider in human behavior is a person's contact rate when they become symptomatic. Do they continue to go about their normal day when showing signs of being sick? This is most likely not the case. Typically, when a person becomes sick they tend to stay at home to get better. When the person first starts to show symptoms they might decide to go home earlier than usual to get more rest. As the symptoms progress and get worse, they will probably stay at home more, reducing the interactions with others. This is represented in the model with Δ *CR* discussed in section 3.3.2. Some diseases' symptoms progress much faster than others. To account for this dynamic change in the severity of different diseases' symptoms, a user could set the Δ *CR* percentage high to represent this dramatic onset of symptoms.

3.1.2. Code Structure

The contact model aims to have the built-in features be able to be used by the user based on the input data enter into the input file. This is done by building the features so they may be turned off or on when needed for a specific disease or in a particular simulation. Another goal is to organize the data and output files produced during and after the simulation in a way so they may be accessed by a user to reproduce the results for future analysis. The type of files used by the simulator are listed in Table 3.2. The name and descriptions of the source code files are listed in Table 3.3.

TABLE 3.2. Type of Simulator Files

File Name	Description
Input File	Contains the disease and population parameters. Figure 3.2 shows an example of how this file is set up.
Output Files	These files are used to store the data produced during the simulation and to plot the graphs of the data.
Code Files	The main code for the simulator is contained in one file while another source file contains the functions for plotting graph using Gnuplot. There are three header files that converts an integer into a string, splits a string into an array, and is used for the Gnuplot source file. Table 3.3 lists the names and descriptions of these files.

TABLE 3.3. Name of Source Code Files

File Name	Description
episimE.cpp	Main source code file where all classes and the *main* function are located.
gnuplot_i.hpp	Header file for the C++ interface to Gnuplot written by Rajarshi Guha.
gnuplot_i.cc	Source code for the C++ interface to Gnuplot written by Rajarshi Guha.
stringexplode.hpp	Header file splits a string into an array of strings for reading in the input.
convert.h	Header file converts an integer value into a string.

Figure 3.1 shows a diagram of the overall code structure. The *episimE.cpp* file reads in the input value then executes the simulation. When the simulation completes, it writes the data to the appropriate output files.

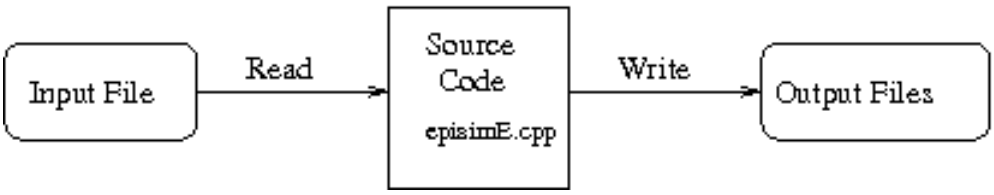

FIGURE 3.1. Code Structure Diagram

3.2. Code Development

A compartmentalized approach was used in developing the code for the contact model. For instance, a problem was approached by simplifying the components in to the most basic form before adding any sophisticated parts. Taking this approach allowed for the development of the code to take on several complexities and layer them to together to create the contact simulator. It also allowed for many features to be added as discussed in section 3.1.1.

The first step was to develop a model for a disease. In order to have both the infectious and symptomatic timelines aligned with each other, a person needed to be asymptomatic while at the same time being in the infectious stage spreading the disease to people they come into contact with. It was important to include both timelines so the model could simulate a person becoming symptomatic thereby knowing when to decrease the contact rate of the individual. Decreasing the contact rate of person when they become symptomatic models a person becoming sick and reducing the amount of people they see in a day. A time unit of one day was used since most diseases are measured by days. This made a person's contact rate the number of individuals they come into contact with per day. However, the simulator may execute in shorter time steps by simply adjusting the time counters, modifying the function that changes the disease state of each person, and by entering the input values to reflect the desired time steps. For example, if hours were wanted instead of days, a latent period of one day would be entered as value of 24 and the contact rate (CR) would be based on how many contacts occurred in one hour.

```
simulation.type = SINGLE_SIMULATION
#string

simulation.seed = 109
#int

disease.latentPeriod = 1
#int

disease.incubationPeriod = 2
#int

disease.infectiousPeriod = 3
#int

disease.infectivity = 0.01
#double

disease.virusLike = true
#bool

population.numberOfPeople = 20000
#long

population.numberOfInfectious = 1
#long

population.numberOfVaccinated = 0
#long

population.percentOfImmune = 0
#double

population.CR = 38
#int

population.DCR = 0
#double

population.resistance = 0.0
#double

population.immuneLevel = 0.0
#double
```

FIGURE 3.2. Sample Input File

3.2.1. Working Model

The typical SIR model did not take into account that an infected individual could be infected but no yet infectious or could be infectious but not yet symptomatic. A fourth disease state was required to account for this possibility so the model could be more dynamic in simulating an outbreak. The utilization of the SEIR model as the foundation for the contact model allowed for the symptomatic period of an infectious disease to be modeled. It also created a way to count the number of exposures of each individual. Analyzing the diagram of the two timelines of a disease showed that an individual needed to have two different time-stamps associated with the current disease state they were in. Since an individual had two timelines, they could be infected but not yet infectious based on the latent and incubation periods of the disease.

When developing an approach to determining how a susceptible person becomes infectious, the simulator used many different methods. The most basic method used to calculate the probability of a person becoming infected was to compare a random number R with the infectivity i. For example, if $R \leq i$, then the susceptible person would become latent. The model then needed a more dynamic approach to take into account a person's number of exposures and resistance level, so an equation using the infectivity i, the resistance r, and exposures ϵ values was derived.

The probability of becoming infected is i, so the probability of becoming infected in ϵ exposures with a resistance level of r is:

$$P = [i * (1 - r)]^{\epsilon} \tag{1}$$

By subtracting 1 from r, Equation 1 represents how much a person is not resistant to the disease. The product of multiplying this result with i is the infectivity of the disease with a person's resistance level. To represent the probability of not becoming infected, $\bar{P}$, in ϵ exposures, the infectivity i is subtracted from hundred percent or 1:

$$\bar{P} = [1 - i * (1 - r)]^{\epsilon} \tag{2}$$

After obtaining the probability of not becoming infected in ϵ exposures with a resistance level of r, subtracting 1 from the result of Equation 2, $\bar{P}$, produces the probability P of becoming infected in ϵ exposures with a resistance level of r:

$$P = 1 - [1 - (i * (1 - r))]^{\epsilon} \tag{3}$$

By developing an equation that incorporated the exposure count, the resistance and immune levels, the simulator's run time was reduced because the model needed to draw less random numbers throughout the entire execution. If the simulator calculated the probability of a person becoming infected every time that person was exposed, there would be a random number needed for every exposure. Using equation 3, this number is reduced to the amount of susceptible individuals who have had contact with an infectious person decreasing the overall runtime of the simulator. Decreasing the runtime of the simulator allows for simulations with large population sizes, which increases the functionality and flexibility of the contact model.

The random experiments use *drand48* of the C++ standard library found in the header file *stdlib.h*. The pseudo-random values returned by *drand48* are non-negative double-precision floating points uniformly distributed between [0.0, 1.0}. Using the probability distribution created by *drand*, we compare a random number with the probability of a person becoming infected. How the random experiments are used is discussed in detail in Chapter 3 subsection 3.2.1.

The seed of the pseudo-random number generator is set based on an input value before the random experiments are conducted. The pseudo-random number generator is seeded with *srand48* at the beginning of each simulation so that the same results may be duplicated when needed. It provides a way to analyze the sensitivity of the simulator to the random numbers and how much they are effected by them. As the experiments show in Chapter 4, the variation induced by different random sequences is small. This shows the simulator can produce similar results regardless of the ordering of the random numbers.

3.3. Code Implementation

At the beginning of the simulation, the input file is read and the values that were read from the input file are set accordingly. Table 3.4 shows the input names and the respective types. The population is generated and initialized and is stored in an array which is the size of the entire population. The values associated with each person are initialized using a class to access and store the data. If the user specifies a percentage of the population that are vaccinated then the simulator randomly assigns a person to be vaccinated. A person who has an immune disease state can be randomly chosen to be vaccinated. In most cases, a person does not know whether they are immune to a disease until after the outbreak has occurred. To model this behavior, a person who is immune to a disease is can be randomly chosen to have a vaccinated disease state.

Once the population array is generated, the data is written to a file to be used throughout the simulation. The primary case or cases of the infectious population is then generated by randomly choosing from the initialized population who have a susceptible disease state and assigning an infectious disease state. It is assumed the primary cases of the infectious population at the beginning of the simulation is just starting the infectious period. To determine whether the infectious population is symptomatic at the start of the simulation, the incubation period is compared to check if it is less than the latent period. If the incubation period is greater than the latent period, primary cases have become symptomatic. There are variables initialized to keep track of the newly susceptible, latent, infectious, and non-infectious individuals of each day throughout the entire simulation.

After the general and infectious populations are set up, the first day of the simulation begins. Before the contacts are created for each infectious individual, the *changeDiseaseStatus* function is called to increment the entire population's disease state time (*timeOfDS*) and to change the disease state if necessary. This function is called before the contacts are generated so if an individual is symptomatic their contact rate will change before making

TABLE 3.4. Inputs

Input Name	Type
simulation.seed	Integer that seeds *drand48*
disease.latentPeriod	Integer value in days
disease.incubationPeriod	Integer value in days
disease.infectiousPeriod	Integer value in days
disease.infectivity	Floating-point value
disease.virusLike	Boolean value
population.numberOfPeople	Long integer
population.numberOfInfectious	Long integer
population.numberOfVaccinated	Long integer
population.percentOfImmune	Floating-point
population.CR	Integer
population.DCR	Floating-point
population.resistance	Floating-point
population.immuneLevel	Floating-point

contact with susceptible individuals. The *changeDiseaseStatus* function uses Algorithm 3.3.1 to determine when individual becomes symptomatic during the course of the simulation.

Algorithm 3.3.1: SYMPTOMATIC($person$)

for each $person \in ipopulation$

do **if** ($diseaseState = latent$ **or** $infectious$) **and** $symptoms = false$

then **if** $timeOfDS > latentPeriod$

then $symptoms = true$

The contacts between susceptible and infectious individuals are generated for the day. The model is only records the contacts that involve at least one infectious individual because the model only needs to keep track of the individuals the infectious population comes into contact with since the infectious individuals is how the disease spreads through a population. This is discussed in detail in section 3.3.2.

At the end of each day, the number of latent and infectious individuals is evaluated. If they are both equal to zero, then there are no more individuals who are infected or infectious, meaning the disease can no longer spread in the population, so the simulation ends. Otherwise, the simulation continues until there are no more latent and infectious individuals. The data produced is then plotted using line graphs. The graphs listed in Table 3.5 are plotted using Gnuplot [13].

TABLE 3.5. Graph Descriptions

File Name	Description
infected_latent_xK.png	Shows the total number of latent and infectious individuals for each day.
newly_numbers_xK.png	Shows the new latent, infectious, non-infectious individuals for each day.

3.3.1. Data Structures

The programming language used to implement the design of the contact model is C++. The simulator uses basic data structures to store and organize the data needed to create a disease and population model. Both the person and disease parameters are stored in separate classes. The class for the population uses the person and disease classes to initialize each individual's parameters and the disease parameters. Figure 3.3 is an example of how the values of the person class is represented. Figure 3.4 is an example of how the disease class is represented. The population class is also where all the functions handling the population are

```
class Person {
    public:
    long id;
    int age;
    DiseaseState diseaseState;
    int timeOfDS;
    int timeOfIncubation;
    bool symptoms;
    int exposureCount;
    int CR;
    double DCR;
    double resistance;
    double immuneLevel;
    long infected;
    Person() { id = 0; }
};
```

FIGURE 3.3. Person Class

located. For example, functions that read and write to the data file to record the population data are in this class. Figure 3.5 is how the population class is represented. To keep track of infectious individuals, a new dynamic array is created. Also in this class the function to generate contacts and change the disease status of a person. Figure 3.6 shows the population array and the dynamic infectious array. When a person x enters the infectious period, a new position is created in the dynamic infectious array that points to the position of the person in the population array. As infectious individuals become non-infectious, their pointer is deleted and the array is adjusted to reflect the new size of the infectious population. The person parameters are accessed by creating a pointer that points from the class to the population

```
class Disease {
  public:
    int latentPeriod;
    int incubationPeriod;
    int infectiousPeriod;
    double infectivity;
    bool virusLike;
    Disease (){}
    Disease (int a,int b, int c, double d,bool e){
        latentPeriod=a;
        incubationPeriod=b;
        infectiousPeriod=c;
        infectivity=d;
        virusLike=e;
    }
```

FIGURE 3.4. Disease Class

array. This is done the same way for the infectious population except with a double pointer. The disease parameter uses a pointer that points from a variable to the class.

3.3.2. Creating Contacts

Since the model is only concerned with the contacts the infectious individuals have, contacts are generated for every infectious person and the rest of the population. This why the infectious population array is important. Instead of going through the entire population array searching for the infectious individuals, the simulator can quickly generate the contacts for the infectious individuals by looping through the infectious population array. The number of contacts for an infectious person depends on the contact rate set in the input file. It also depends on if the infectious individual is symptomatic. If they are symptomatic, their contact

```
class Population{
    Disease *disease;
    Person *population;
    Person **infectiousPopulation;
  public:
    Population()
    ~Population()
    void setDisease(latent, incubation, infectious, infectivity)
    void writeDisease()
    void readDisease()
    void generatePopulation(size, percentOfImmune)
    void initPopulation()
    void readPopulation()
    void infectPeople(numOfInfectious)
    void writePopulationToFile()
    void makeContacts(day)
    void infectPerson(x)
    void removeDisease(x)
    void changeDiseaseStatus()
    int calculateAndPrintNumbers(day, fileName)
    void writeNumbersToFile()
    void singleSimulation()
    void plotLatentInfectious()
    void plotNewlyNumbers()
};
```

FIGURE 3.5. Population Class

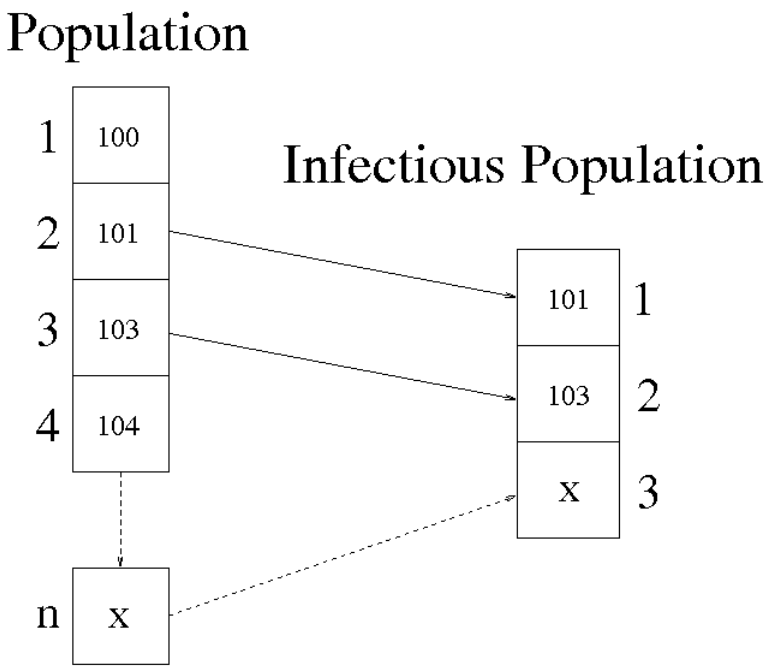

FIGURE 3.6. Array Data Structure Diagram

rate decreases by the Δ contact rate percentage. For every day they are symptomatic their contact rate will decrease accordingly.

To simplify how individuals come into contact with an infectious individual, a susceptible individual is randomly selected to be one contact for the infectious individual. The susceptible person's exposure count is then increment. At the end of the day, after all the infectious population's contacts are created, the susceptible individuals with more than one exposure are tested in a random experiment to determine if they become infected and enter the latent period. Please refer to Algorithm 3.3.2 on how this is done. A susceptible person can be randomly selected more than once. This means they may be exposed to multiple infectious individuals several times in one day increasing their chances of becoming infected.

Algorithm 3.3.2: INFECT PERSON($exposureCount$)

for each $person \in population$

do
- **if** $exposureCount > 0$ **and** $person.diseaseState = susceptible$
- **then**
  - $probability = 1 - (1 - (infectivity * (1 - resistance)))^{\epsilon}$
  - **if** $drand48() <= probability$
  - **then**
    - $person.diseaseState = latent$
    - $person.timeOfDS = 0$
    - $newlyLatent++$

CHAPTER 4

EXPERIMENTS

4.1. Analysis of Results

A line graph that shows the total number of infectious individuals and the total number of latent individuals for each day during the outbreak reveals several different types of information. Typically when an epidemic occurs, this line graph shows a curve that contains one peak which occurs near the middle of the graph. This is called the epidemic curve. The epidemic curve gives a simple visual representation of the outbreak's magnitude and time trend. This can be used to help predict how a disease's different time periods effect the overall spread of the disease.

In the following analysis of the output data produced by the simulator, the experiments will show how one input value can effect the curve of an epidemic. In all the experiments in this section, the influenza-like virus will be used to create a baseline model to analyze the data. Once an individual becomes infected by a particular virus strain, they can no longer become infected again. To model this the *virusLike* input is always set to *true* so that all individuals become immune to the disease when they reach the recovered state. Also, for simplicity of the experiments the *resistance* and the *immuneLevel* inputs will always be set to 0, not reducing a susceptible individual's probability of becoming infected. To remain consistent in every experiment, the seed in every experiment will be set to 1 unless otherwise specified.

When analyzing the infectious and latent curves of a simulation, to help determine if an epidemic occurred, it is important to determine what percentage of the population was infected. To calculate this by looking at the curve, the following formula can be used:

$$\frac{1}{IP} \cdot \int I(t) \tag{4}$$

Where *IP* is the infectious period and *I(t)* is the number of infectious individuals at time *t*. Using the peak value of the total infectious curve will give the total number of infected individuals at time *t*.

4.1.1. Random Seeds

Multiple experiments were executed with different seeds and the same input parameters. This was done to show the simulation results are independent of the random number sequence to produce consistent output. It is important for the model to not be dependent on the sequence of random numbers because the simulator needs to produce consistent results regardless of the random number sequence. This allows the simulator to be executed on multiple platforms and produce similar results.

Figure 4.1 shows 5 different experiments, each with a different seed using the input values listed in Table 4.1. The overall shape of the curve shows that the experiments are quantitatively very similar. Even though the curves may be shifted or higher than others, figure 4.2 shows the shape curve is still maintained when averaging the values of all the experiments. With a population of 20,000 people, close to 30% of the population was infected resulting an epidemic lasting 189 days. This approximates the reported incidence levels and epidemic time frame for the annual influenza cycle.

When analyzing the epidemic curve, several values that are representative of the curve can help determine the similarities between graphs. These values include the length of the curve (total days), the total number of infected individuals, the height of the curve (infectious peak), when the peak occurred (peak day), and the days when 5% and 95% of the infected population was reached. Table 4.2 lists the values and their standard deviation. There is little deviation between all the different seeds executed in Experiment 1 resulting in consistent output regardless of what random sequence is used.

4.1.2. Contact Rate

The contact rate of each individual affects the rate at which an infectious disease spreads throughout a population. Sometimes if the contact rate is low, the disease will infected

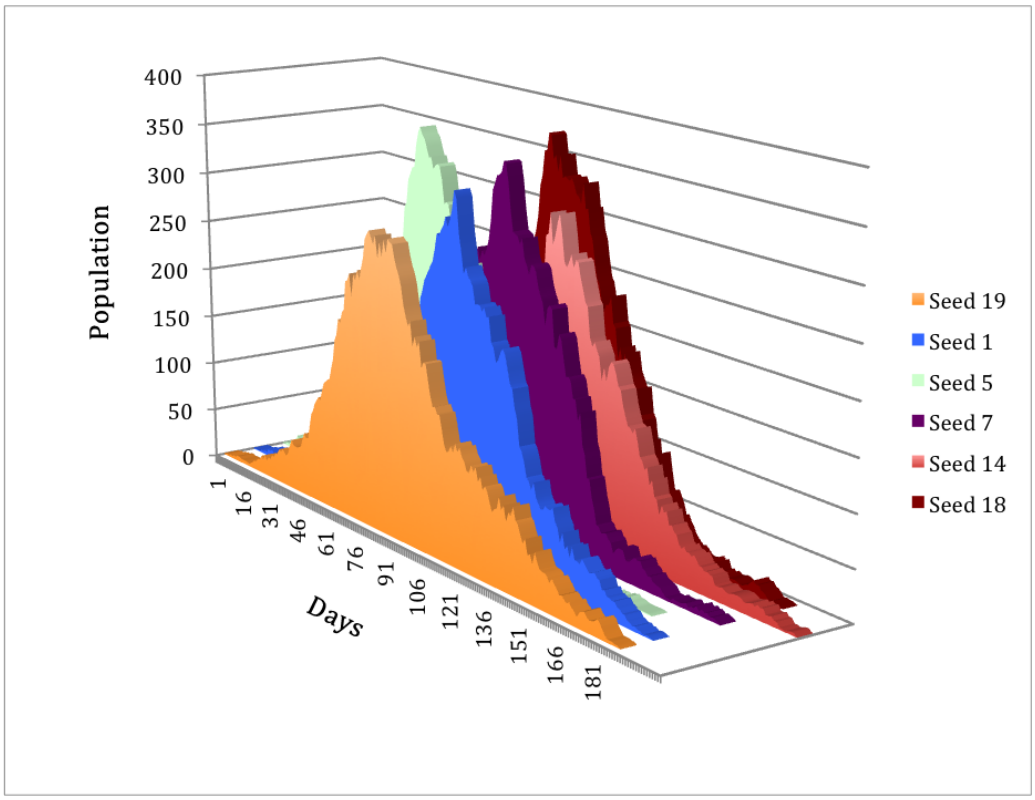

FIGURE 4.1. Experiment 1: Different Seeds

TABLE 4.1. Input Values for Experiment 1

Input	Value
disease.latentPeriod	1
disease.incubationPeriod	2
disease.infectiousPeriod	3
disease.infectivity	0.01
population.numberOfPeople	20000
population.numberOfInfectious	1
population.numberOfVaccinated	0
population.percentOfImmune	0
population.CR	40
population.DCR	0

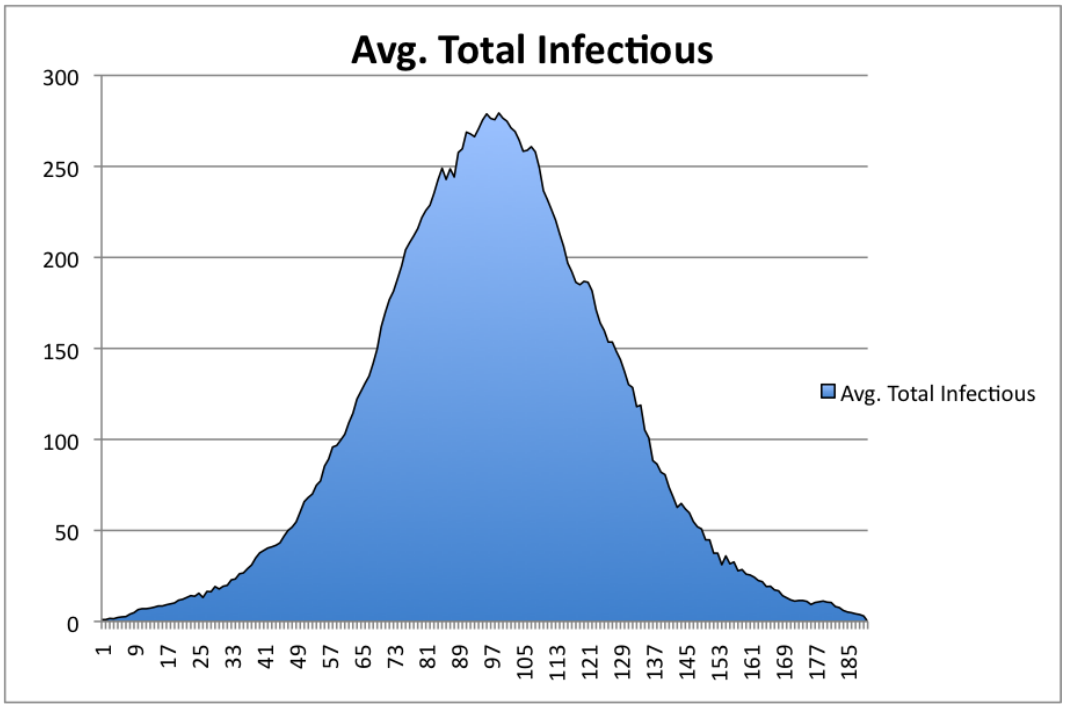

FIGURE 4.2. Average of Experiment 1

TABLE 4.2. Experiment 1 Representative Values of Epi Curve

Seed	Total Days	Total Infected	Infectious Peak	Peak Day	Day 5% Infected	Day 95% Infected
1	189	5972	340	107	59	154
5	176	6261	356	77	41	123
7	190	6407	356	101	52	139
14	207	6065	309	112	65	155
18	190	6796	372	94	54	142
19	189	6167	289	91	49	151
σ	9.14	269.75	30.12	11.45	7.54	11.11

only a small percentage of the population not resulting in a epidemic. Figure 4.3 shows five simulations with a different contact rate (CR) for each execution. Table 4.3 shows the input values used for this simulation. As the contact rate (CR) value decreased, the number of

infected individuals also decreased. This is because the spread of the disease is dependent on the number of contacts an infectious individual has during the outbreak.

The overall result of an epidemic is sensitive to changing the contact rate, so changing the *DCR* value will also effect the epidemic result. In Experiment 3, three different *DCR* values were used to show how it effects the total number of infected individuals. Figure 4.4 shows when increasing the *DCR* input value, the number of individuals infected is reduced. Reducing an infectious individual's contact rate once they become symptomatic reduces the spread of the disease. This causes the total number of infected individuals to be reduced.

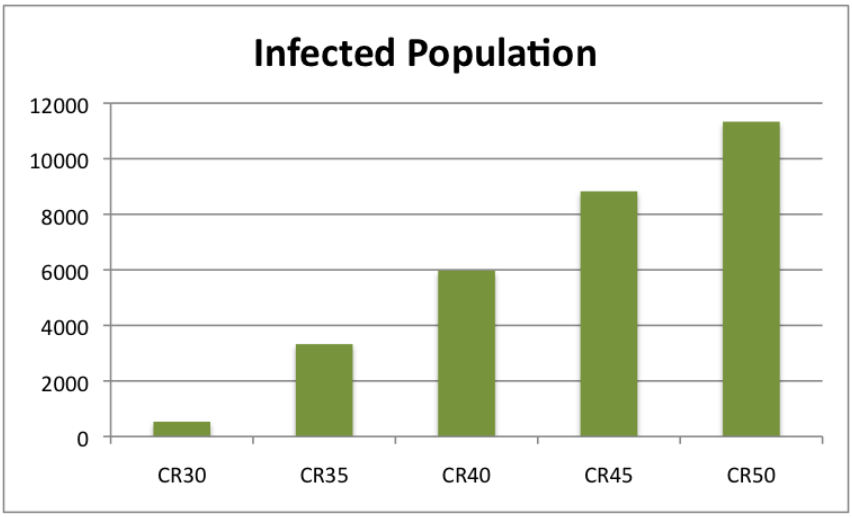

FIGURE 4.3. Experiment 2

4.1.3. Latent Period

When increasing the latent period, the overall duration of the outbreak will also increase. This is because it takes longer for an individual to become infectious reducing the rate at which the disease spreads. If the latent period is longer than the incubation period, then the contact rate of a person will decrease before they become infectious. This results in the disease spreading at a more reduced rate than if the latent period was less than or equal to the incubation period. As the disease spreads at a reduced rate, the probability of more individuals becoming infected increases. In Experiment 4, a simulation where the latent

TABLE 4.3. Input Values for Experiment 2

Input	Value
disease.latentPeriod	1
disease.incubationPeriod	2
disease.infectiousPeriod	3
disease.infectivity	0.01
population.numberOfPeople	20000
population.numberOfInfectious	1
population.numberOfVaccinated	0
population.percentOfImmune	0
population.CR	see graph
population.DCR	0

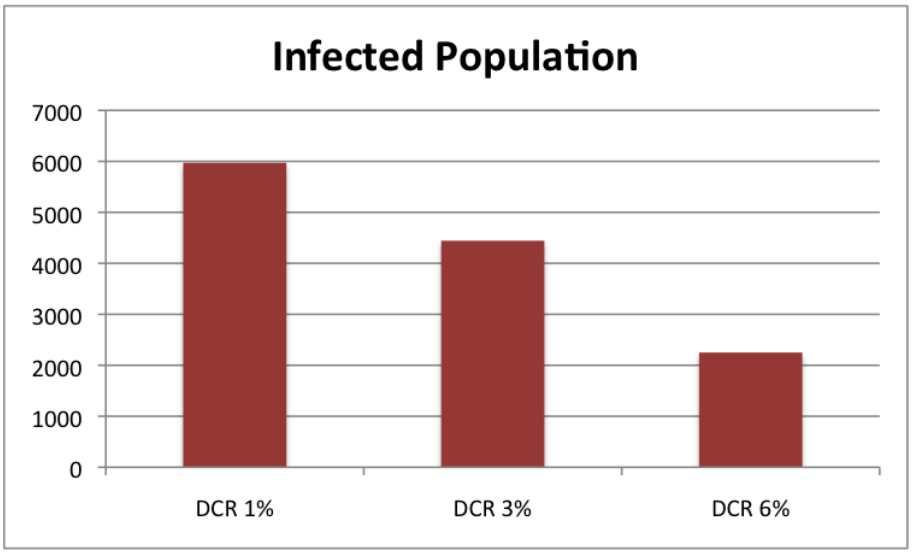

FIGURE 4.4. Experiment 3

period is longer than the incubation period resulted in an outbreak lasting 271 days. When compared to the baseline average epidemic curve in Figure 4.2, this is sufficiently longer.

Looking at Table 4.10, the values of Experiment 4 are larger than the average of Experiment 1 because the epidemic curve is longer. For example, the infectious peak is greater

TABLE 4.4. Input Values for Experiment 3

Input	Value
disease.latentPeriod	1
disease.incubationPeriod	2
disease.infectiousPeriod	3
disease.infectivity	0.01
population.numberOfPeople	20000
population.numberOfInfectious	1
population.numberOfVaccinated	0
population.percentOfImmune	0
population.CR	40
population.DCR	see graph

because more people were infected while the slow rate of the disease outbreak cause the peak to occur later. 40% of the population was infected during the simulation, an increase from Experiment 1. By analyzing Figure 4.5, it shows as the total number of latent and infectious increase, the two lines remain separated by 3 days and in Experiment 1 they are separated by 1 day. This shift in the graphs reflect the disease timeline because the latent period occurs before the infectious period.

4.1.4. Incubation Period

The incubation period of a disease effects when an individual's contact rate will begin to decrease causing either the spread of the disease to be reduced or increased during the simulation. This effects the total number of infected individuals and the height of the epidemic curve. As seen in Experiment 3, reducing the contact rate of an individual causes the rate at which the disease spreads to decrease. So when reducing a person's contact rate during the outbreak by setting the DCR value greater than 0, the epidemic curve is shorter in the length and the peak.

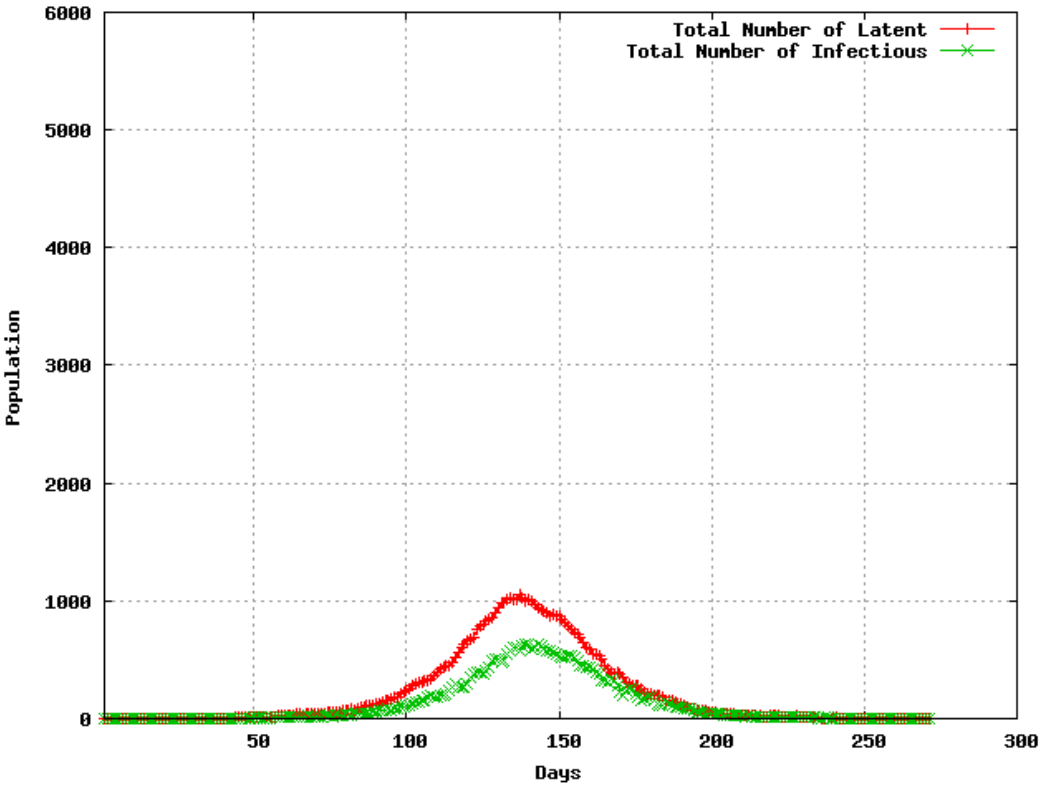

FIGURE 4.5. Experiment 4

TABLE 4.5. Input Values for Experiment 4

Input	Value
disease.latentPeriod	3
disease.incubationPeriod	2
disease.infectiousPeriod	3
disease.infectivity	0.01
population.numberOfPeople	20000
population.numberOfInfectious	1
population.numberOfVaccinated	0
population.percentOfImmune	0
population.CR	40
population.DCR	0

In Experiment 5, a simulation was executed with an incubation period of 4 days. In this simulation, 35% of the population became infected. This shows that by increasing the incubation period greater than the latent period, the simulation will result in a shorter outbreak with fewer individuals of the population becoming infected. Figure 4.6 shows an epidemic curve lasting 150 days with a smaller number of individuals becoming infected than in Experiment 4. Looking at Table 4.10, the total number of infected for Experiment 5 is less than Experiment 4, while the peak of Experiment 5 occurred earlier because of the shortened outbreak of the disease. The length of the epidemic curve in Experiment 5, the total number of days, is over 40% less than the length in Experiment 4.

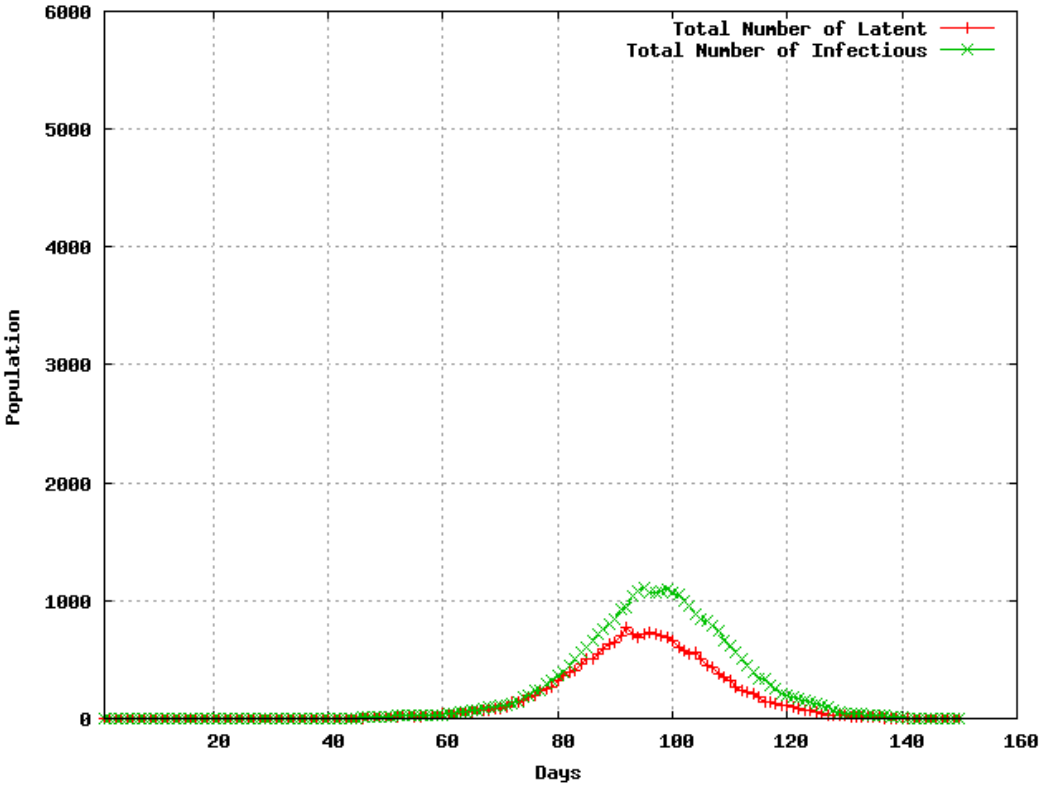

FIGURE 4.6. Experiment 5

4.1.5. Infectious Period

The infectious period directly effects the amount of individuals infected by the disease. With a long infectious period and a low *DCR* value, there are more chances for a susceptible

TABLE 4.6. Input Values for Experiment 5

Input	Value
disease.latentPeriod	1
disease.incubationPeriod	4
disease.infectiousPeriod	3
disease.infectivity	0.01
population.numberOfPeople	20000
population.numberOfInfectious	1
population.numberOfVaccinated	0
population.percentOfImmune	0
population.CR	40
population.DCR	0.25

person coming into contact with an infectious individual, increasing the percentage of the population becoming infected. In Experiment 6, a simulation with a DCR value of 0 was executed. Figure 4.7 shows 88% of the population was infected. This is a very high percentage of the population becoming infected with only a infectivity of 1%. This figure also shows the epidemic curve only lasting just over 100 days, shorter than any experiment thus far. When having a high infectious period, infected individuals have more time to come into contact with susceptible individuals, allowing the disease to spread at a high rate. Experiment 6 shows that having a high infectious period will effect the rate at which the disease spreads, similar to how the contact rate can effect the rate of the outbreak.

To show how the DCR value will only effect the height of the epidemic curve with a high infectious period, a simulation similar to Experiment 6 was executed in Experiment 7. The difference between these two experiments was the contact rate of a person during the outbreak was decreased by setting the DCR value greater than 0 to 6%. This causes the total number of infected individuals to decreases, affecting the height of the epidemic curve. Figure 4.8 shows 78% of the population was infected, less than in Experiment 6. In

Table 4.10, the values for Experiment 6 and 7 are very close to each other except for the total number of infected value and the infectious peak, which is the height of the curve.

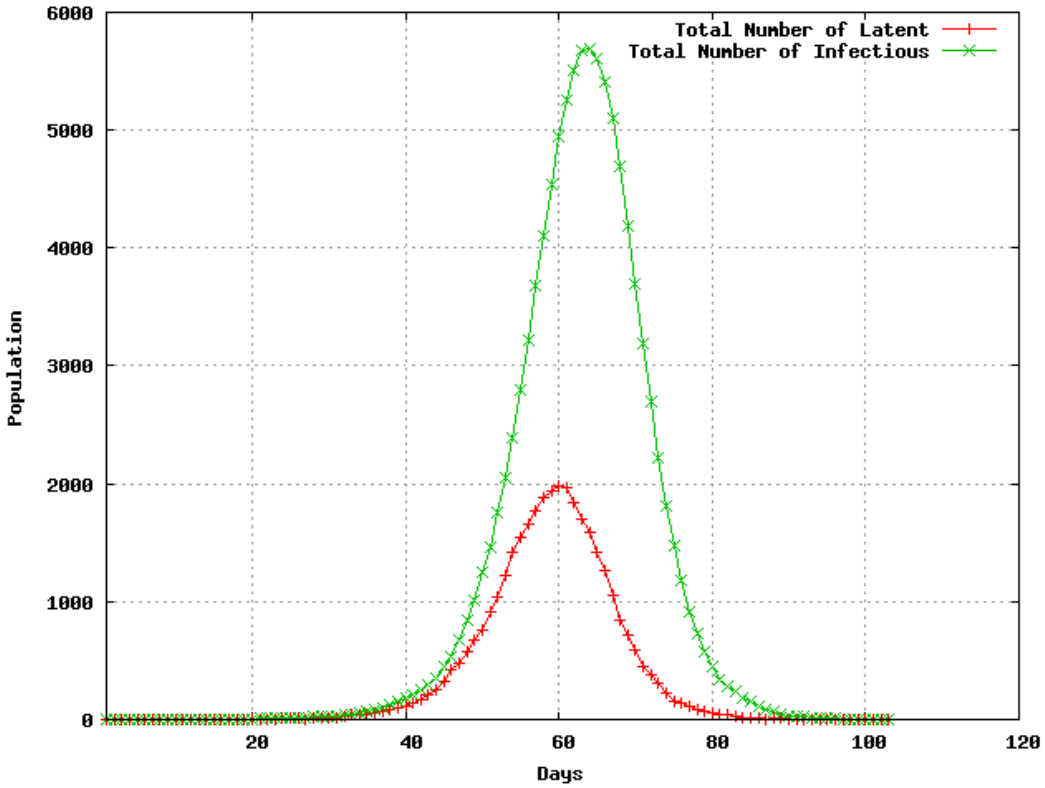

FIGURE 4.7. Experiment 6

4.1.6. Infectivity

When increasing the infectivity, the probability of a susceptible person who has been exposed becoming infected increases, resulting in an increase in the total number of infected individuals, raising the height of the curve. Since the disease infects more people in the population, it spreads at a high rate resulting in a short outbreak. In Experiment 8, a simulation with an infectivity of 10% was executed. Figure 4.9 shows over 90% of the population was infected during this simulation that lasted 52 days.

This experiment resulted in the largest percentage of the population becoming infected. It was also the shortest epidemic curve. The outbreak is short because of the random sequence. By changing the random sequence so that only a small amount of individuals

TABLE 4.7. Input Values for Experiment 6

Input	Value
disease.latentPeriod	1
disease.incubationPeriod	2
disease.infectiousPeriod	6
disease.infectivity	0.01
population.numberOfPeople	20000
population.numberOfInfectious	1
population.numberOfVaccinated	0
population.percentOfImmune	0
population.CR	40
population.DCR	0

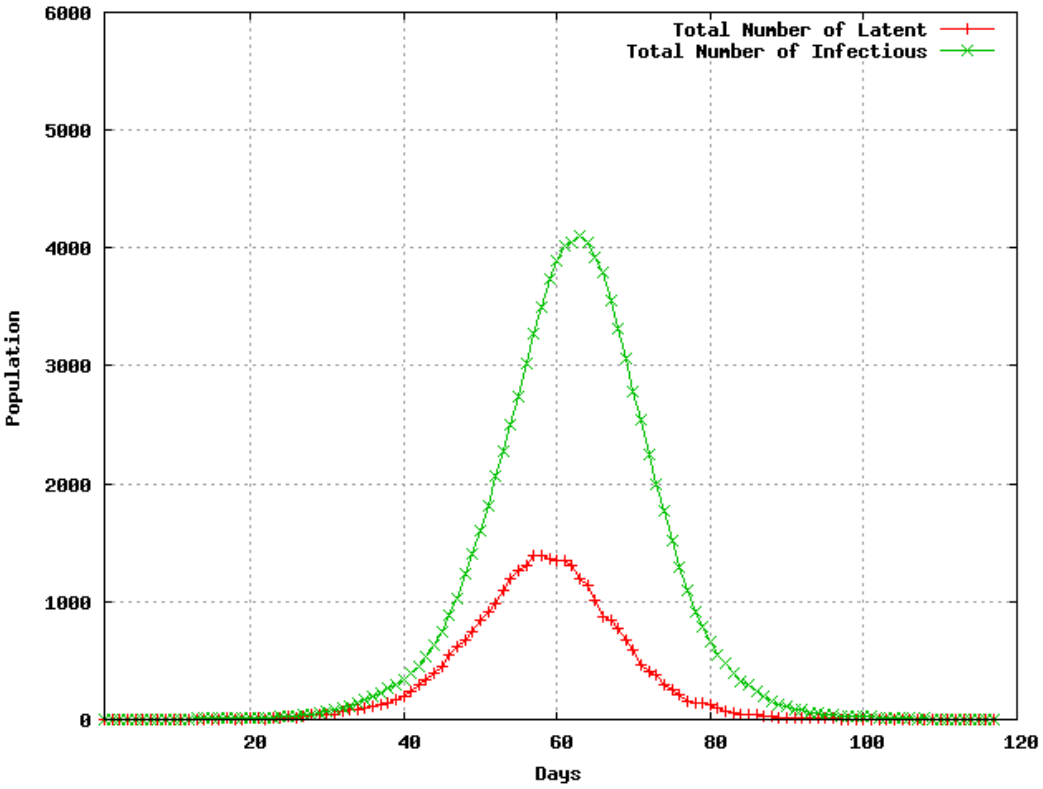

FIGURE 4.8. Experiment 7

TABLE 4.8. Input Values for Experiment 7

Input	Value
disease.latentPeriod	1
disease.incubationPeriod	2
disease.infectiousPeriod	6
disease.infectivity	0.01
population.numberOfPeople	20000
population.numberOfInfectious	1
population.numberOfVaccinated	0
population.percentOfImmune	0
population.CR	40
population.DCR	0.06

become infected, the simulator can produce results that are similar to having a low infectivity. This simulation demonstrates that the infectivity of a virus is crucial to how fast it spreads in a population and how many overall individuals are infected. Looking at the values in Table 4.10, the peak day occurs the earliest of all the experiments because of the length of the epidemic curve resulting in the days when 5% and 95% of the infected population was reached to occur the earliest. The total amount of days the outbreak took, or the length of the curve, is sufficiently smaller than any other experiment, showing that a random sequence was generated so that the majority of the population was infected.

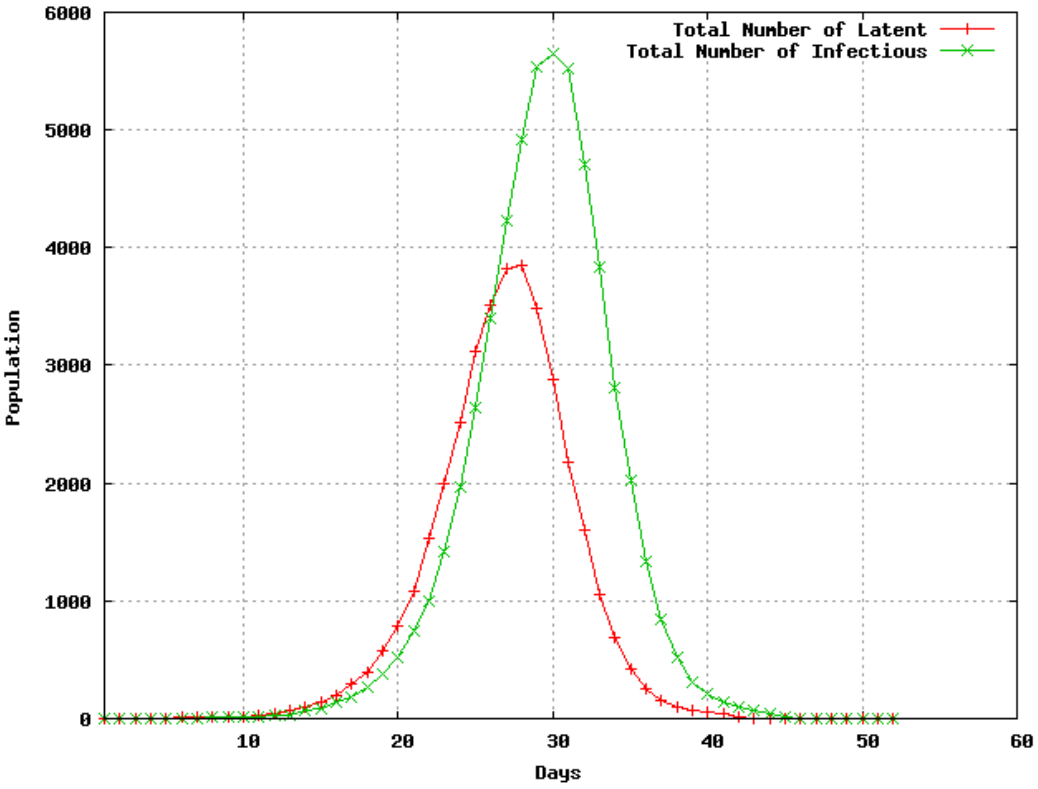

FIGURE 4.9. Experiment 8

TABLE 4.9. Input Values for Experiment 8

Input	Value
disease.latentPeriod	1
disease.incubationPeriod	2
disease.infectiousPeriod	3
disease.infectivity	0.1
population.numberOfPeople	20000
population.numberOfInfectious	1
population.numberOfVaccinated	0
population.percentOfImmune	0
population.CR	40
population.DCR	0

TABLE 4.10. Representative Values of Epi Curve

Experiment	Total Days	Total Infected	Infectious Peak	Peak Day	Day 5% Infected	Day 95% Infected
1	189	6261	279	98	36	154
4	271	11522	629	140	100	188
5	150	11168	1106	95	75	120
6	103	17608	5694	64	48	78
7	117	15842	4103	63	48	81
8	52	18578	5647	30	22	38

CHAPTER 5

CONCLUSION

5.1. Summary

The limitations of traditional models based on the contact rate of individuals creates the demand for new computational models in studying the complexity of the spread of infectious diseases in the world population. The contact model addresses this demand for new computational tools for the analysis of outbreaks in a population. This contact model is the base model for the spread of an infectious disease based on the population's individual contacts. It can be used for planning interventions schemes and assisting in the readiness of preparedness plans throughout the world.

The simulator provides the analysis of the contact model's implementation. It demonstrates how the contact model can be utilized by epidemiologist and public health officials for developing effective planning, prevention, and intervention programs. The contact model's design features are a contributing factor to providing detailed information on what disease and population characteristics effect the spread of an infectious disease based on the contacts between infectious individuals and the susceptible population. By analyzing the results produced in Chapter 4, the contact model contributes to the effectiveness of the simulator in computational epidemiology. The contact model showed how the interactions of individuals when modeling an infectious disease is important for predicting epidemics.

The contact model demonstrates several new approaches that can be used in developing a simulator that models multiple infectious diseases. By allowing the features of the simulator to be used based on the input of the user, the contact model creates flexibility without having to modify the code. This is important so the simulator can be used in multiple applications allowing a wide range of users to benefit from the simulator.

It is essential to have the overall structure of the contact model based on the SEIR model approach to incorporate the disease and symptomatic timelines of an infectious disease so the decrease in the contact rate can be accurately calculated. Having the function to decrease the contact rate of an individual based on whether they are symptomatic or asymptomatic, increases the usability of the contact model. This approach contributes to the main functionality of the contact model.

At the moment, even though the contact model uses a Monte-Carlo approach to randomly distribute the contacts between individuals, it can still be used by epidemiologist and health professionals to determine the effects of a particular disease characteristic on the overall epidemic in relation to the contact rate of individuals in the population. The experiments conducted in Chapter 4 were limited to showing how each input parameter effected the epidemic curve. Further simulations could explore the robustness of contact model under different conditions. For example, many more random seeds could be tested over multiple platforms. Another useful way to further exploration of research of robustness would be to create a new baseline model to conduct experiments with.

The results of the experiments suggest the contact rate of each individual and the disease and symptomatic timelines play critical role in determining how an outbreak of an infectious disease spreads. If epidemiologist and health officials are to create effective planning, prevention, and surveillance programs, these two factors must play a major role in the decision process.

5.1.1. Future Work

The contact model is by far not a comprehensive model for the spread of infectious diseases. It was designed as a starting point for future work in providing a in-depth contact model to epidemiologist. There are plenty of features that will be added to this model. For instance, instead of having a fixed number for the latent, incubation, and infectious periods, to simulate closer to what happens in the real world, an approach to distributing a range of each period for each individual based on gender, age, and geographic location can be

developed to more accurately simulate an outbreak. Also, creating a distribution for the contact rate of individuals based on these characteristics would result in a more realistic model with a wide range of applications.

For a more realistic model, there needs to be a more accurate distribution system for creating the contacts between indiviudals. This could be done using a network of clusters. Since individuals of a population tend to come in contact with a specific cluster of people, the best way to approaching this problem would be to create clusters of people for each infectious individual.

A better approach to creating and initializing the population would use census data of a city's population to simulate an outbreak in a particular city or area of the country, instead of randomly assigning values to each individual. This could provide an accurate prediction of how an infectious disease might spread in a specific population. Epidemiologist then could determine what kinds of prevention and intervention procedures need be taken in that area.

The contact model is valuable to tool analyzing how the characteristics of an infectious disease and a population will have on the overall result of an epidemic. With the new features, the contact model will prove to be a more valuable tool for epidemiologist while studying the progression of infectious disease, thereby allowing them to utilizes public health resources at an optimal level.

BIBLIOGRAPHY

[1] Frank Ball and Peter Neal, *A general model for stochastic sir epidemics with two levels of mixing*, Mathematical Biosciences 180 (2002), 73–102.

[2] F. Brauer and P. van den Driessche, *Models for transmission of disease with immigration of infectives*, Mathematical Biosciences 171 (2001), 143–154.

[3] Fred Brauer, *Some simple epidemic models*, Math. Biosci. Eng. 3 (2006), no. 1, 1–15.

[4] S. Busenberg and P. van den Driessche, *Analysis of a disease transmission model with varying population size*, J. Math. Biol. 29 (1990), 257.

[5] Linda L. Hill, Scott J. Crosier, Terence R. Smith, and Michael Goodchild, *A content standard for computational models*, D-Lib Magazine 7 (2001), no. 6.

[6] Edwin D. Kilbourne, *Influenza pandemics of the 20th century*, Emerg Infect Dis 12 (2006), no. 1.

[7] A.M. Law and W.D. Kelton, *Simulation modeling and analysis*, 3rd ed., McGraw-Hill, Boston, MA, 2000.

[8] S. A. Levin, B. Grenfell, A. Hastings, and A. S. Perelson, *Mathematical and computational challenges in population biology and ecosystems science*, Science 275 (1997), 334–343.

[9] S. Mounier-Jack and R. Coker, *How prepared is europe for pandemic influenza? analysis of national plans*, Lancet 367 (2006), 1405–1411.

[10] Kenrad. E. Nelson, Carolyn. M. Williams, and Neil. M. H. Graham, *Infetious disease epidemiology: Theory and practice*, Aspen Publications, Inc. , Gaithersburg, Maryland, 2001.

[11] G. Rose, *Epidemiology*, The Oxford Medical Companion (1994), 244–248.

[12] D. Simth, *Predictability and preparedness in influenza control*, Science 312 (2006), 392–394.

[13] T. Williams, C. Kelley, R. Lang, D. Kotz, J. Campbell, G. Elber, A. Woo, and many others, *Gnuplot*, http://www.gnuplot.info, March 2008.