July 2019

A Computational Simulation Model for Predicting Infectious Disease Spread using the Evolving Contact Network Algorithm

Buyannemekh Munkhbat

A COMPUTATIONAL SIMULATION MODEL FOR PREDICTING INFECTIOUS DISEASE SPREAD USING THE EVOLVING CONTACT NETWORK ALGORITHM

A Thesis Presented

by

BUYANNEMEKH MUNKHBAT

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN
INDUSTRIAL ENGINEERING AND OPERATIONS RESEARCH

May 2019

Mechanical and Industrial Engineering

A COMPUTATIONAL SIMULATION MODEL FOR PREDICTING INFECTIOUS DISEASE SPREAD USING THE EVOLVING CONTACT NETWORK ALGORITHM

A Thesis Presented

by

BUYANNEMEKH MUNKHBAT

Approved as to style and content by:

_____
Chaitra Gopalappa, Chair

_____
Hari Balasubramanian, Member

_____
Shannon Roberts, Member

_____
Sundar Krishnamurty, Department Head
Mechanical and Industrial Engineering

DEDICATION

To my family

ACKNOWLEDGEMENTS

I want to thank my advisor and mentor, Professor Chaitra Gopalappa. I am deeply grateful for her continuous support, guidance, and encouragement for me to advance both academically and professionally. I appreciate her generosity and patience with me during my graduate studies. I want to extend my gratitude for my committee members, Professor Hari Balasubramanian and Professor Shannon Roberts. I truly appreciate and value their useful feedback and critical inputs that assisted me in making improvements in my work.

I gratefully acknowledge the funding I received towards my master's degree from National Institute Of Allergy And Infectious Diseases of the National Institutes of Health.

I also greatly appreciate the experience I received through the research internship at the Stanford Graduate School of Business during my graduate studies – many thanks to Dr. Vitor Hadad for his mentorship and support.

My sincere appreciation goes out to my friends and Disease Modeling lab mates for their support, suggestions, and friendship throughout the years. I want to thank Vijeta Deshpande for being the best lab mate and his wonderful friendship during our graduate studies. I am forever grateful to my dearest friend Angela Upreti for her continuous support, and I am thankful that I met her while I was at UMass Amherst. Also, I would like to express my gratitude towards Javkhlan Bold and his family, one and only Mongolian family in Amherst, for their love and care for me during my graduate years. I would also like to thank Temuge Enkhbaatar for his support, encouragement, and love over the years.

Finally, I want to say a heartfelt thank you to my parents, my grandparents, my little sister, my cousins and my relatives for their unconditional love and support.

ABSTRACT

A COMPUTATIONAL SIMULATION MODEL FOR PREDICTING INFECTIOUS DISEASE SPREAD USING THE EVOLVING CONTACT NETWORK ALGORITHM

MAY 2019

BUYANNEMEKH MUNKHBAT, B.A., MOUNT HOLYOKE COLLEGE

M.S., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Chaitra Gopalappa

Commonly used simulation models for predicting outbreaks of re-emerging infectious diseases (EIDs) take an individual-level or a population-level approach to modeling contact dynamics. These approaches are a trade-off between the ability to incorporate individual-level dynamics and computational efficiency. Agent-based network models (ABNM) use an individual-level approach by simulating the entire population and its contact structure, which increases the ability of adding detailed individual-level characteristics. However, as this method is computationally expensive, ABNMs use scaled-down versions of the full population, which are unsuitable for low prevalence diseases as the number of infected cases would become negligible during scaling-down. Compartmental models use differential equations to simulate population-level features, which is computationally inexpensive and can model full-scale populations. However, as the compartmental model framework assumes random mixing between people, it is not suitable for diseases where the underlying contact structures are a significant feature of disease epidemiology. Therefore, current methods are unsuitable for simulating diseases that have low prevalence and where the contact structures are significant.

The conceptual framework for a new simulation method, Evolving Contact Network Algorithm (ECNA), was recently proposed to address the above gap. The ECNA combines the attributes of ABNM and compartmental modeling. It generates a contact network of only infected persons and their immediate contacts, and evolves the network as new persons become infected.

The conceptual framework of the ECNA is promising for application to diseases with low prevalence and where contact structures are significant. This thesis develops and tests different algorithms to advance the computational capabilities of the ECNA and its flexibility to model different network settings. These features are key components that determine the feasibility of ECNA for application to disease prediction. Results indicate that the ECNA is nearly 20 times faster than ABNM when simulating a population of size 150,000 and flexible for modeling networks with two contact layers and communities. Considering uncertainties in epidemiological features and origin of future EIDs, there is a significant need for a computationally efficient method that is suitable for analyses of a range of potential EIDs at a global scale. This work holds promise towards the development of such a model.

TABLE OF CONTENTS

Page

ACKNOWLEDGEMENTS v

ABSTRACT vi

LIST OF TABLES xi

LIST OF FIGURES xii

CHAPTER

1. INTRODUCTION 1

2. BACKGROUND 5

2.1 Epidemic Models 5

2.2 Computational Models 7

2.3 Current Gaps and Research Question 8

2.4 Graph Theory 11

2.5 Graph Representation 12

2.5.1 Adjacency Matrix 12

2.5.2 Adjacency List 13

2.5.3 Comparison 13

2.6 Graph Properties 13

2.7 Basic network generating models 15

2.7.1 Random graphs 16

2.7.2 Configuration model 17

2.7.3 Chung-Lu model 18

2.7.4 Exponential Random Graph Models 18

2.7.5 Preferential Attachment 19

3. METHODS 21

3.1 Notations 22

3.2 Algorithm 23

3.3 Empirical Validation of the Algorithm 26

3.4 Baseline ABNM Model 28

3.5 Models 28
3.5.1 Model 1.0: Multi-contact Evolving Network using Adjacency Matrix 29
3.5.1.1 Base ABNM 30
3.5.1.2 Objective of the model 31
3.5.1.3 Model parameters 32
3.5.1.4 The data structure, complexity, and technology 32
3.5.2 Model 1.1: Multi-contact Evolving Network using Adjacency List 33
3.5.2.1 Base ABNM 33
3.5.2.2 Objective of the model 34
3.5.2.3 Model parameters 34
3.5.2.4 The data structure, complexity, and technology 34
3.5.3 Model 2.0: Multi-communities Evolving Network using *NetworkX* and Preferential Attachment 36
3.5.3.1 Base ABNM 37
3.5.3.2 Objective of the model 40
3.5.3.3 Model parameters 40
3.5.3.4 The data structure, complexity, and technology 40

4. RESULTS 42
4.1 Model 1.0 42
4.1.1 Epidemic projections and network properties 44
4.1.1.1 Prevalence 44
4.1.1.2 Incidence 45
4.1.1.3 Network Properties 46
4.1.1.4 Programming debugging 47
4.1.2 Computation Time 48
4.1.3 Graph Visualizations 49
4.2 Model 1.1 51
4.2.1 Epidemic projection and network property 51
4.2.1.1 Prevalence 51
4.2.1.2 Network Properties 52
4.2.2 Computation Time 53
4.3 Model 2.0 54
4.3.1 Epidemic projection 55
4.3.2 Graph Visualization 56

5. CONCLUSIONS 58
5.1 Summary 58
5.2 Discussion 59

5.3 Limitations 61
5.4 Conclusion 62

6. SUPPLEMENT MATERIALS ……64

APPENDIX

THE SOFTWARE CODE FOR DIFFERENT IMPLEMENTATIONS OF GRAPH OBJECT……....65

BIBLIOGRAPHY 68

LIST OF TABLES

Table Page

Table 2-1: Space and time complexities of representations 13
Table 2-2: Basic generalization of network model 15
Table 3-1: The Evolving Contact Network Algorithm 24
Table 3-2: Model 1.0 Parameters 32
Table 3-3: Model 1.0 Algorithm 33
Table 3-4: Java HashSet complexity 34
Table 3-5: ECNA algorithm using Adjacency List 35
Table 3-6: Finding eligible contacts using Adjacency List 35
Table 3-7: Model 2.0 Parameters 40
Table 3-8: Model 2.0 Algorithm 41
Table 4-1: Network properties of different networks that are used in Model 1.0 43

LIST OF FIGURES

Figure Page

Figure 2-1: The spread of the tuberculosis. Image from (Andre et al. 2011) 8

Figure 2-2: An undirected random graph..... 12

Figure 2-3: A directed random graph..... 12

Figure 2-4: Configuration model stub matching mechanism 18

Figure 3-1: A schematic representation of the ECNA with $\boldsymbol{N = 12, n = 2, k = 2}$..... 25

Figure 3-2: Eligible contacts for newly infected..... 26

Figure 3-3: Eligible edges for newly infected..... 26

Figure 3-4: Validation pipeline of the ECNA. It first collects data, which varies depending on the model, from ABNM then generate the evolving network while simulating the epidemic using the ECNA and compare the results with results from ABNM..... 27

Figure 3-5: Multi-contact network..... 29

Figure 3-6: Contact types describe the different types of interactions among agents. The dashed lines emphasize that the graphs have the same nodes, but the edges are distinct… 30

Figure 3-7: NetLogo graphical interface, where a number of input values can be entered, for an ABNM. Red lines represent family contacts, and blue lines represent friend contacts. 31

Figure 3-8: Generating a network with two groups where each group is scale-free from two independent scale-free networks by removing edges from each network then adding edges between networks. 38

Figure 3-9: A final network, the baseline network for ANBM, with two groups where each group is scale-free..... 39

Figure 3-10: Degree distributions of the generated network in Figure 4-9 before and after rewiring the edges 39

Figure 4-1: Epidemic projection comparison of the ECNA (Model 1.0), ABNM and compartmental model. 5th, 50th, and 95th percentiles of 100 simulations. 45

Figure 4-2: Incidence projection comparison of the ECNA (Model 1.0) and ABNM. 5^{th} and 95^{th} percentiles of 100 simulations..... 46

Figure 4-3: Model 1.0 the convergence of average degrees of friend's contact 47

Figure 4-4: Program debugging in MATLAB code 48

Figure 4-5: Computation time comparison of ECNA (Model 1.0) and ABNM 49

Figure 4-6: An evolution of a network with two-contact type using the Evolving Contact Network Algorithm. a) An infected contact network at the initial stage of the outbreak b) An infected contact network where the epidemic spread over. 50

Figure 4-7: Epidemic projection comparison of the ECNA (Model 1.1), ABNM and compartmental model. 5th, 50th, and 95th percentiles of 100 simulations. 52

Figure 4-8: Model 1.1 the convergence of average degrees of friend's contact 52

Figure 4-9: Computation time comparison of ECNA (Model 1.0), ECNA (Model 1.1) and ABNM (NetLogo)..... 53

Figure 4-10: Computation time comparison of ECNA (Model 1.1) and ABNM (NetLogo) 54

Figure 4-11: Epidemic projection comparison of the ECNA (Model 2.0) and ABNM. 5th, 50th, and 95th percentiles of 10 simulations. 55

Figure 4-12: An evolution of network with two-communities using the Evolving Contact Network Algorithm. a) An infected contact network at the initial stage of the outbreak b) An infected contact network where the epidemic spread over. Light blue and pink colors represent uninfected persons, deep blue and red colors represent infected persons in the network. 57

CHAPTER 1

1 INTRODUCTION

As there are occurrences of new and re-emerging infectious disease outbreaks, there is a need for tools to predict dynamics as accurate as possible and as efficient as possible so that public health officials can implement optimal intervention methods at the initial stages of outbreaks. Simulation models offer such tools for estimating the characteristics of a specific disease outbreak. There exist different models, such as *compartmental* or *agent-based*, for estimating the spread of the disease for diseases with different disease dynamics.

The traditional differential-equation-based *compartmental model* was first introduced by Kermack and McKendrick (Kermack and McKendrick 1927), and this model forms the basis of modern quantitative epidemiology. Compartmental models are based on compartmentalization of individuals based on their disease status and the transmission between states are defined by differential equations (Anderson and May 1991; Diekmann and Heesterbeek 2000). Though this model tracks the changes in compartments of individuals, it does not specify which individual was involved within the compartment. The basic compartmental model is the susceptible-infectious-recovered (SIR) model, and all compartmental models follow the non-stationary Markov processes structure and assume that the host population is homogeneously mixed. There exist different compartmental model structures depending on the disease characteristics. For example, the SIR model is most suitable for diseases that confer lifelong immunity, such as measles, and the susceptible-infectious (SI) model is suitable for diseases that do not have treatment at the moment such as human immunodeficiency virus (HIV) and Ebola virus (Keeling and

Eames 2005). While these models allow us to gain insight into disease transmission process and to study threshold quantities such as basic reproduction number R_0 (Paulinevan den Driessche 2017), they do not model a contact-network structure of the human network to study disease transmission (Simon, Taylor, and Kiss 2011).

Although the homogeneously mixed equation-based compartmental model is suitable for simulating the spread of highly contagious diseases that are easily spread at large scale, it creates prediction error when it applies to diseases that have a lower number of daily contacts or in highly clustered population (Smieszek, Fiebig, and Scholz 2009). Real-world human networks tend to be highly clustered, and the spread of infectious diseases that are transmitted through *close-contact* such as Ebola or HIV depends on heterogenous mixing within the population. This mixing takes numerous individual information such as population size and density (Suryaprasad et al. 2013), the age structure of the population (Merli and Hertog 2010), the composition of household (Adams 2016; Cauchemez et al. 2009), and demographic and cultural practices (Alexander et al. 2015) into account. Therefore, it is important to incorporate different communities or groups based within the network and their mixing between the groups when simulating such diseases where the community structure is important.

Agent-based network models (ABNM) are well suited to handle these individual-level complexities by focusing on the interactions among agents. ABNM can represent modeling of disease spread in a realistic contact network, and it simulates persons at the individual level, this gives the flexibility to model specific close contact network. However, this extra complexity of ABNM models significantly increases computational requirements

and often requires scaling down the actual population size due to limited time and resources (Rahmandad and Sterman 2008; Goodreau et al. 2012).

Current models use these extreme simulation techniques that have a trade-off between increased modeling capacity and computational time complexity. ABNM is problematic for low prevalence diseases as the number of infected persons becomes negligible or vanishes when scaling-down the population size. Further, as ABNM generates the full population contact structure at the start of the simulation, it is impractical to use for simulating disease spread at a global scale in real-time decision-making environments. Thus, there is no suitable method that can model the spread of diseases that transmit through close contact and have a low prevalence or are widespread geographically.

The conceptual framework for a new simulation technique Evolving Contact Network Algorithm (ECNA) was recently proposed to address this computational challenge of simulating diseases with low prevalence (Eden et al. 2018). This thesis presents an empirical analysis of the ECNA to test its accuracy, computational efficiency, and flexibility to different network types and population settings. The ECNA integrates *individual-level* modeling capacity of agent-based network models for simulating infected individuals and contacts, with computation efficiency of compartmental models for simulating uninfected contacts at *population-level*. During disease transmission, the social contacts between susceptible and infected persons are significant, whereas contacts between uninfected persons are not significant. The overview of this algorithm is building a contact network as people become infected at each simulation step by generating only infected persons and their close contacts. The main advantage of this algorithm is

computational efficiency when simulating disease outbreak with low prevalence in a large population.

Chapter 2 presents a literature review on epidemic models, current research gaps and research objective as well as a technical background that is needed for the algorithm development. Chapter 3 describes the algorithm and the models that were developed for this study. Chapter 4 then presents the results of the models on the accuracy, computational efficiency, and the flexibility of the algorithm. Chapter 5 discusses limitations and conclusions of the study.

CHAPTER 2

2 BACKGROUND

2.1 Epidemic Models

Epidemic models are powerful tools that help to understand and predicting infectious disease transmissions. With the ever-changing history of infectious disease types and patterns, the effective control system and predictive modeling of infectious disease have been rapidly improving (Hethcote 1994; Cohen 2000). Remarkable progress has been made in population-level compartmental models that incorporate homogeneous mixing within each subpopulation. For example, the Global Epidemic and Mobility (GLEaM) has been used to access international travel restrictions during 2009 influenza and 2014-2016 Ebola outbreak (Tizzoni et al. 2012; Bajardi et al. 2011; Poletto et al. 2014; Balcan et al. 2010). These population-level models divide the population into compartments based on their disease state and assume homogenous mixing between contacts (Ajelli et al. 2010). This assumption is suitable for highly infectious diseases like flu, measles, or dengue fever (Coburn, Wagner, and Blower 2009; Derouich, Boutayeb, and Twizell 2003). Not only the compartmental models with homogenous mixing assumptions are not able to model a relationship between individuals, but also they overestimate the number of infections of the diseases that transmit through *close-contact* (Drake et al. 2015).

Agent-based models used to simulate at individual-level and these models will provide a more accurate epidemic prediction for diseases where the contact structure is important. Siettos et al. applied an agent-based model with small-world network structure assumption to model 2014 Ebola outbreak in Liberia and Sierra Leone, and their estimate best fitted the Ebola outbreak data reported by WHO (Siettos et al. 2015). This model

showed a strong argument for modeling diseases that transmit through close contact using an agent-based model.

Moreover, Willem et al. did a systematic review of agent-based models for infectious disease publications that were published between 2006 and 2015 (Willem et al. 2017). They filtered and reviewed 698 papers, and they found that agent-based modeling application to infectious disease model is increasing each year (38 to 115 from 2006 to 2015). They noted that most papers among the selected papers are on agent-based modeling for close-contact diseases (27%), followed by influenza (23%). This study concluded that there is an availability of individual-level data as well as rising interest in precision modeling (Willem et al. 2017).

The pattern of epidemics, which transmits diseases from one person to another, is determined by not only the disease characteristics such as its infectiousness and recovery rate but also by the network structures within the population (Potterat et al. 2002; Keeling 2005; Rocha, Liljeros, and Holme 2011). The act of disease spreading is one kind of dynamic process that takes place on networks, and this process is often referred as *cascading behavior* or *social contagion* (Bauch and Galvani 2013; Jiang et al. 2014). Studying characteristic patterns of a structure at the network level helps to facilitate infectious disease spreading, particularly ones that transmit through close-contact.

For many agent-based social network simulation models, an underlying social network – the collections of social ties among friends or family – is required and this social network can be represented as a graph (Newman 2006; Rahmandad and Sterman 2008; Hamill and Gilbert 2009).

2.2 Computational Models

Eubank et al., one of the pioneers of applying social network to epidemic modeling, explored dynamic bipartite graphs to model movements of individuals between specific locations (Eubank et al. 2004). They have built an agent-based simulation tool EpiSims that combines realistic estimates of population mobility with parameterized models for simulating the progress of disease within an agent and of transmission between agents (Eubank et al. 2004). They found that contact network among people is clustered, but the locations graph is scale-free from their case study in smallpox spread in Portland, Oregon. Eubank et al. concluded that a scale-free locations graph suggest that efficient outbreak detection system can be done by placing sensors in locations with high degrees and targeted vaccination could be more effective than mass vaccination during epidemics. EpiSims simulates the disease spread on the network after producing the social networks and people's movement with the help of TRANSIMS, the transportation analysis system that produces estimates of a social network based on transportation infrastructures (Eubank et al. 2004), whereas the ECNA proposes to generating the social network as the disease spread on the network.

There are simulation models including EpiSimdemics, which is a more advanced version of EpiSims, that use a scalable parallel algorithm to simulate the diseases in a large population at individual levels (Barrett et al. 2008; Ferguson et al. 2003; Longini 2005). However, the common challenges of these models are limited by supercomputer storages that deals with a large amount of social network data and this could be a bottleneck during an outbreak in remote places.

Infectious disease modeling at individual-level in a smaller population with realistic social network information researches have been done using patient contact tracing

methods (Read, Eames, and Edmunds 2008; Mossong et al. 2008; Le Polain de Waroux, Oliver et al. 2018). For example, Andre et al. reinforced this analogy and examined Tuberculosis (TB) contact investigation procedures during the outbreak (Andre et al. 2011). They collected TB patient data and traced the close contacts of the patients by interviewing them (Figure 2-1). Willem et al. show the network of TB patients and their contact (Willem et al. 2017). They concluded that a network-informed approach helped to focus on TB control much effectively and helped to analyze the disease spread.

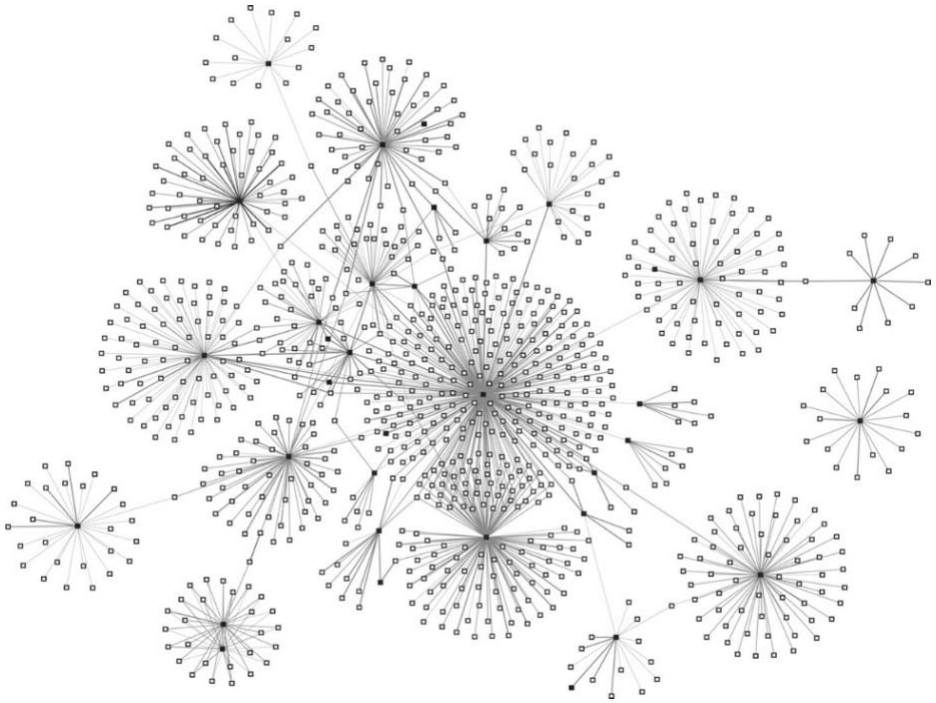


Figure 2-1: The spread of the tuberculosis. Image from (Andre et al. 2011)

2.3 Current Gaps and Research Question

There have been an increasing number of agent-based models in epidemic spreading with the help of increasing computational power, availability of specific data, and an awareness of the limitations of homogenous mixing models (Bansal Shweta, Grenfell Bryan T, and Meyers Lauren Ancel 2007; Enright and Kao 2018). However, these

models often require a full contact network information, which needs high computational power, before simulation the epidemic over (Volz et al. 2011; Siettos et al. 2015) or scaling down the population into a smaller sample, which will also scale down the final prevalence.

A model that requires full contact information is may not be suitable during the event of an outbreak where rapid epidemic forecasting is needed for public health decision making. Siettos et al. investigated the epidemic dynamics of Ebola Virus Disease in Liberia and Sierra Leone using an agent-based model whose dynamics evolve on small-world networks where its size matches the demographics of each country (C. Siettos et al. 2015). Though this model estimated the incidence with high accuracy, the model required to generate a contact network with millions of nodes and simulated the infection over the full network. This requirement makes the model computationally expensive and the computation time increases when population size increases.

The population size is often scaled down to computationally feasible size to avoid the high computational cost of modeling the entire population. Although this technique saves computational cost, this would result in estimation error when it is applied to an epidemic with low prevalence. For example, the prevalence of the 2014 Ebola outbreak in three West African countries Guinea, Liberia, and Sierra Leone yielded to be 0.12 percent. This prevalence was computed based on the total cumulative cases of 28,616 (Center for Disease Control 2016) and the total population size of 23.28 million of three countries ("Data for Sierra Leone, Liberia, Guinea | Data" n.d.). Therefore, simulating 100,000 persons, representative of the three countries, will yield a total of 120 infected cases of the virus. Considering the infection number is 120 at the peak of the epidemic, we would not have a sample that is statistically significant to simulate at the initial stages of the outbreak.

Therefore, agent-based models face the challenge of simulating diseases with low prevalence in a large population at a lower computational cost.

Agent-based contact network models often generalize contact types into a single layer contact by averaging degree, clustering coefficients or other network structures (Danon et al. 2011). Another challenge in agent-based modeling for infectious disease is a rarity of the availability of reusable open-source code for these computational models. However, ComplexNetworkSim ("Welcome to ComplexNetworkSim's Documentation! — ComplexNetworkSim v0.1.2 Documentation" n.d.) in Python package, EpiModel and SimInf in R (Widgren et al. 2016) allow to simulate disease on a simple contact network, which is generated using already existing random graph generators, rather than from explicitly specified network contacts (Enright and Kao 2018).

The Evolving Contact Network Algorithm (ECNA) is proposed to fill these gaps and aims to generate the contact network while simultaneously simulating the epidemic which results in similar infection prediction as simulating over a full-network at a lower computational cost. This thesis focuses on the formulation and empirical validation of the ECNA and its implementation that results in computational efficiency over existing agent-based models that require the full network before simulating the infection. The ECNA generates the contact network while simulating the disease and allows us to consider multiple contact types between individuals and multiple communities within the population.

2.4 Graph Theory

A ***simple graph*** $G(V, E)$ consists of a non-empty finite set $V = \{v_1, v_2 \dots, v_n\}$ of n elements called ***node*** where $|V| = n,\ n > 0$ and a finite set $E = \{e_1, e_2, \dots, e_m\}$ of m distinct pairs of distinct elements of V called ***edge*** where $E \subseteq V \times V,\ |E| = m,\ m \geq 0,$ $e_k = (v_i, v_j),\ \forall v_i, v_j \in V, i \neq j$ (Wilson 1998).

In a social network, a node in a graph represents a person, and an edge between two nodes represents a relationship between two persons that would allow for disease transmission. In graph theory, edges can represent directional interaction between two nodes such that there are ***undirected*** and ***directed*** graphs (Newman 2010). Figure 2-2 and Figure 2-3 show undirected and directed random graphs. From an epidemiological point of view, the direction of a graph is essential since it possesses information and restrictions on how the disease transmits. In this paper, we assume disease can transfer between any connected close individuals. Therefore, we are concerned with generating ***simple undirected graphs***, i.e., no ***loops*** or ***multiple edges*** are allowed since we assume each susceptible person will be prone to infected from any of its infected contacts.

A ***giant component*** is a fully connected component that contains a finite fraction of the entire graph's nodes (Newman 2010). If a disease starts in the giant component, the prevalence of the disease increases with the network size, while if the disease starts outside of the giant component, the total number of infected people will be limited. In our case, we are interested in simulating a disease that starts in a giant component in which it needs a rapid projection of the epidemic and makes decisions.

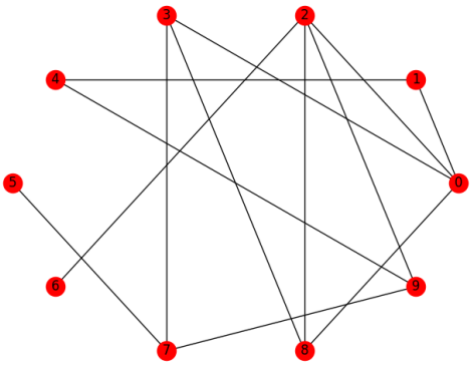


Figure 2-2: An undirected random graph

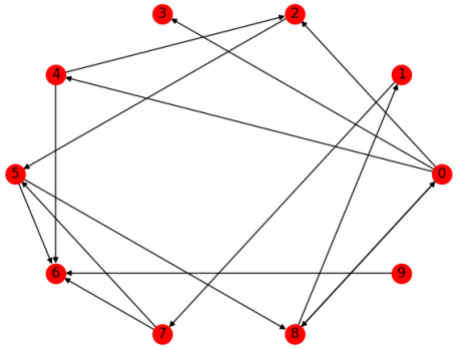


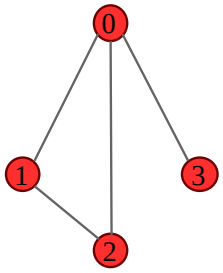
Figure 2-3: A directed random graph

2.5 Graph Representation

There are two main graph representations: the adjacency matrix and the adjacency list (Newman 2010).

2.5.1 Adjacency Matrix

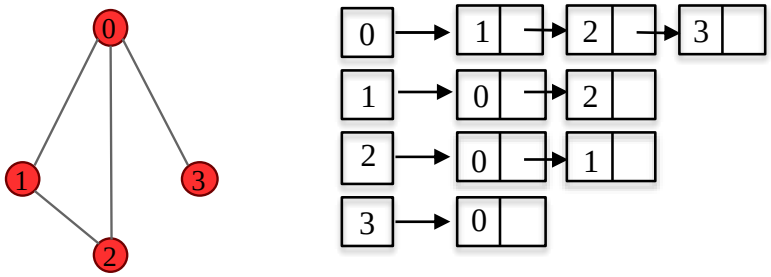
An adjacency matrix is a square $|V| \times |V|$ matrix A with elements a_{ij} is 1 if node i and j are connected and zero otherwise. The elements of this square matrix describe if a pair of nodes are adjacent (connected) or not in the graph. If a graph is undirected, the adjacency matrix is symmetric.



	0	**1**	**2**	**3**
0	0	1	1	1
1	1	0	1	0
2	1	1	0	0
3	1	0	0	0

2.5.2 Adjacency List

A more space-efficient way to implement a graph is an adjacency list that is a collection of unordered lists. Each list corresponds to a node v and contains the set of adjacent nodes of v in the graph. An adjacency list representation is more compact for a graph is sparsely connected.



2.5.3 Comparison

It is vital to understand the trade-offs between two graph representations before implementing algorithms and models that are based on graphs. Table 2-1 contains the space and time complexities of the Adjacency Matrix and Adjacency List representations.

Table 2-1: Space and time complexities of representations

	Space	Checking if (v_i, v_j) is an edge	Identifying all edges
Adjacency Matrix	$\Theta(\|V\|^2)$	$\Theta(1)$	$\Theta(\|V\|^2)$
Adjacency List	$\Theta(V+E)$	$O(\deg(v_i)$	$\Theta(V+E)$

2.6 Graph Properties

There is a large number of graph properties that have been defined to characterize different aspects of the complex networks. The ECNA focuses on simulating the diseases

where the spread of infection is not random but via close-contact., i.e., the more infected contacts, the more chances of being infected. Therefore, the ***degree*** and ***clustering coefficient*** properties of a graph, which contain such direct contact information among other graph metrics, are used to evaluate a graph that is generated by ECNA.

The number of infected people at each time step is used for validation of epidemiological property. The average degree infected population in the network are used for comparison of network properties with other existing models.

- **Degree**: The d number of edges that are originated from node i in an undirected network is the degree of node i, and we write it as d_i, i.e., $k_i = \sum_j a_{ij}$. The average degree of the network $\langle k \rangle$ is the average of the value k_i over all nodes in the network.
- **Degree distribution**: In the undirected network, the degree distribution $P(k)$ represents the probability that a random node has degree k.
- **Degree correlation:** Two-node degree correlation can be measured by means of the conditional probability $P(k'|k)$ that an edge from a node of degree k is connected to a vertex of degree k'.
- **Clustering coefficient:** The clustering coefficient c_i of node i is defined as the ratio between the number of existing triads that is originated at node i, and the number of all possible such triangles at node i. The average clustering coefficient of the network $\langle c \rangle$ is the average of the value c_i over all nodes in the network.

2.7 Basic network generating models

There exist different real-world network structures that are characterized by variability in their graph metrics and statistical properties. The network is not limited by social network, but it includes information network, food webs, as well as citation networks, neural networks and more. Therefore, the existence of network classification has motivated a theoretical research effort in the field of studying different network generation models (Pastor-Satorras et al. 2015). The basic and broad generalization of these models that are reviewed in this section is in Table 2-2. Plus, exponential *random graph model* (ERGM) and *preferential attachment* are discussed.

As a real-world human contact network tends to be highly clustered and the number of contacts of a person is dependent on the person, the graphs with clusters and dependent edges are needed to be used as a base model to validate the ECNA. Also, it is observed that a network of human sexual contacts is scale-free that its degree distribution follows power-law with an exponent between 2 and 3 (Liljeros et al. 2001; Barabási, Ravasz, and Vicsek 2001; Schneeberger et al. 2004).

In this study, we developed models to generate random graphs with high clustering and non-random graphs without clustering using the ECNA. We adopted the configuration model technique for generating random graphs and the preferential attachment model technique for generating non-random graphs in the algorithm.

Table 2-2: Basic generalization of network model

	Independent edges	Dependent edges
Identical nodes	Random graph: $G(n,p)$	Random graph: $G(n,m)$
Non-identical nodes	Chung-Lu model	Configuration model

2.7.1 Random graphs

The most basic probabilistic network model is called the random graph or sometimes referred to the Erdős–Rényi random graph (Paul Erdős, Alfréd Rényi 1960). This graph generating model is typically denoted $G(n,p)$ and the model starts with n nodes and p the probability that an edge $e = (v_i, v_j)$ exists, for all $v_i, v_j \in V$, and this creates a random graph with approximately $\binom{n}{2}p$ edges. Therefore, the average degree of a node is $\langle k \rangle = (n-1)p$. Here the degree distribution is in binomial form, and binomial distribution approaches the Poisson distribution $P(k) = e^{-\langle k \rangle}\frac{\langle k \rangle^k}{k!}$ When the network is large ($n \to \infty$).

Alternatively, $G(n,m)$ random graph takes a fixed n number of nodes and generates m number of edges with equal probability. The average degree of a network is $\langle k \rangle = \frac{2m}{n}$. Classical random graphs have Poisson distributions, which has a rapid decay because of the large factorial in the denominator. But the degree distributions of real-world networks decay much slower (Sergey Dorogovtsev 2010).

The clustering coefficient of a random graph decays to zero in the limit of a large graph. The calculation of the clustering coefficient is derived from the following:

$$\langle c \rangle = \frac{(number\ of\ triangles)}{(number\ of\ connected\ triples)} \propto \frac{\binom{n}{3}p^3}{\binom{n}{3}p^2} = p = \frac{\langle k \rangle}{n-1}$$

where $\langle k \rangle$ is the desired average degree of a network and the average degree of an individual will be negligible compared to the total population in the large network.

All nodes in random graphs are *iid* because all nodes have the same chance of being selected to link with one another, but edges in $G(n,p)$ are independent while edges in $G(n,m)$ are dependent because of a limited total number of edges.

2.7.2 Configuration model

As it was shown in empirical studies, most real-world networks follow power-law distribution while a random graph follows a Poisson distribution when the graph is sparse. One way of improving this aspect of the random graph is by using a model called the configuration model *(Bender and Canfield 1978; Molloy and Reed 1995)*. This model takes a fixed degree distribution as an input to construct the network in contrast to a traditional random graph takes a fixed average degree as an input.

Its construction is as follows:

Each node is pre-assigned to the degree that is drawn from a given degree distribution $P(k)$, subject to the conditions $m \leq k_i \leq N$, where m is the desired minimum degree and $\sum_i k_i$ is an even number. The reason why the total number of degrees in a network is an even number is that we randomly match a pair of nodes by their pre-assigned "stubs" together. Thus, a random graph with any given degree distribution can be constructed with this model by taking a uniformly matching on the "stubs" attached to nodes. Figure 2-4 shows a simple representation of the construction of the configuration model on a graph of N=6 with each node has stubs of 2, 4, 2, 1, 3, 5 and its "stubs-matching" using configuration model. After each stub is linked, the degree distribution is still preserved.

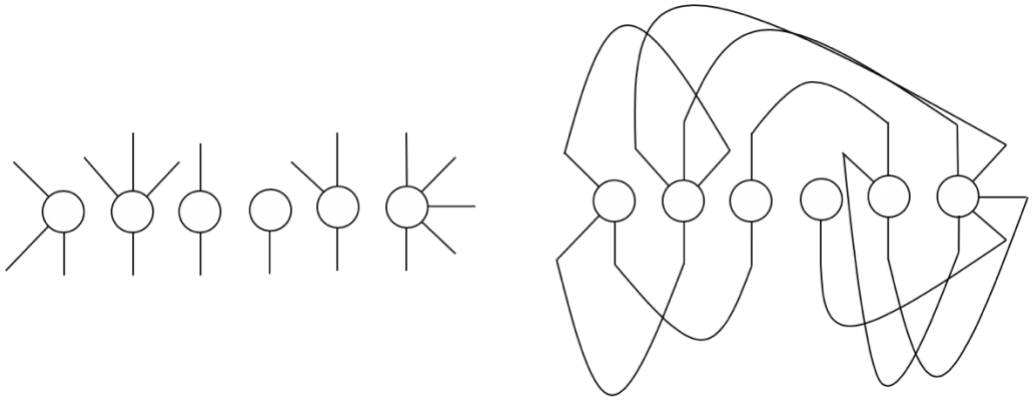


Figure 2-4: Configuration model stub matching mechanism

2.7.3 Chung-Lu model

The Chung-Lu model is a random graph model that is most closely related to the configuration model (Chung and Lu 2002). Instead of being generated by a fixed degree sequence like configuration model, the Chung-Lu model is parametrized by $w = (w_1, \dots, w_n)$ and $w_i > 0$ where w_i is an expected degree of i. The model correctly samples graphs with a given degree sequence for most well-behaved degree sequence.

2.7.4 Exponential Random Graph Models

The Exponential Random Graph Models (ERGM), also known as 'p* models,' are useful for generating networks with its network properties are close to a given set of properties. This model allows the user to generate a network based on which network property the user is concerned more.

It suggests that even though networks could evolve into different structural realizations, they should have some basic features in common.

This kind of common feature concept is called a statistical ensemble of network, $\mathcal{G} = \{G\}$, plus probability distribution $P(G)$, over $\mathcal{G}$. Here, $P(G) \propto e^{H(G)}$, that is exponential in the

so-called graph Hamiltonian, $H(G)$, which determines various network properties within the ensemble.

2.7.5 Preferential Attachment

The randomly connected Erdős–Rényi (ER), or reconnected Watts-Strogatz (WS) models not only do not represent real networks which follows a power-law degree, but also fail to incorporate two key features of real networks: growth and preferential connectivity. Those static networks provide a good approximation when the properties of the dynamical processes evolve faster than the structure of the network changes. In traditional epidemic models have applied static network models to provide predictive analytics on epidemics under assumptions of the diseases are highly infectious and the host population is homogenously mixed and fully susceptible. However, the class of growing network has been useful for modeling epidemics in a non-homogenous network.

The Barabasi-Albert (BA) power-law preferential attachment model allows creating a network with power-law distribution (Barabasi and Albert 1999). This model differs from the configuration model by its growth characteristics, in which nodes and links are added over time.

BA model follows a rule that newly added edges will tend in general to be connected to nodes chosen via some preferential attachment. The simplest of these rules are defined as follows:

i) It starts with a small number m_0 of connected nodes, and introduce a new node with $m(\leq m_0)$ edges that link the new node to m different nodes that are already present in the system at every time step to incorporate the growing feature of a real network.

ii) In contrast to the random network models, this model incorporates preferential connectivity by choosing node i to link to the new node with probability $\prod(k_i) = \frac{k_i}{\sum_{j=1}^{N} k_j}$, where k_i is the current degree of i. Thus, the nodes that have higher degree will have a higher chances of getting linked to more nodes.

This model evolves into a scale-free network that has k edges following a power-law with an exponent $\gamma = 3$ (i.e., $(P(k) \sim k^{-3})$ (Barabasi and Albert 1999; Sergey Dorogovtsev 2010).

Different from random graphs, non-random graphs including scale-free networks provide a degree-correlation of neighbors information (Fotouhi and Rabbat 2013). This information is especially crucial for ECNA (when it generates a scale-free network) since the degree of an infected person, and its contacts degree should be available as soon as the person added to the network. The conditional degree distribution, $P(L = l|k)$ where probability distribution of L given specific degree k, determines the distribution of the degrees of all neighbors of a node of degree k. Fotouhi and Rabbat studied the conditional degree of scale-free networks and presented the analytical model (Fotouhi and Rabbat 2013). Based on the previous study (Eden et al., *in review*), developed an alternative numerical model to estimate the conditional degree of scale-free networks using a non-linear neural network.

CHAPTER 3

3 METHODS

The Evolving Contact Network Algorithm (ECNA) generates the contact network of only infected persons and their immediate contacts, as such, evolves as more people become infected (Eden et al. 2018). It maintains the network properties at each time step when more nodes are added to the network.

A real contact network is a combination of different contact types and communities. This thesis focuses on the validation of the algorithm and implementation of its application to different models where different network structures are considered. Among various validation techniques, a combination of *animation* and *comparison to other models* techniques was used to validate the models that use ECNA to simulate epidemic. The *animation* technique provides model results graphically during the simulation run, and *comparison to other models* compare the proposed model result with other existing models (Sargent 2010).

The objective of the ECNA is obtaining comparable accuracy over traditional computational agent-based models while minimizing space and time consumption when simulating diseases that have a low prevalence. Specifically targeting for a low prevalence disease requires the ECNA have increased accuracy in the early stage of the epidemic are comparable to the projections that were produced by traditional agent-based models.

3.1 Notations

- Contact types and population

Ω	A set of contact types that are not random but static, e.g., household family and social contacts
k	A contact type. $k \in \Omega$
N	Population size
n	Number of initially infected nodes
M_t	The number of people in the network at time t; $M_t \leq N$

- Adjacency Matrix representation of contact types

$\mathcal{A}_{k,t}$	A binary matrix for a contact type k of size $N \times N$ at time t. Here only $M_t x M_t$ matrix will have information at time t, and the rest of the matrix is zeros because they will not be generated yet. (Expanding matrix size at each t is computationally more expensive than having fixed size.)
$a_{k,ij}$	An element of $\mathcal{A}_{k,t}$ in row i and column j, then $a_{k,ij} = 1$, if i and j are contacts, and 0 otherwise.

- Infection status

$\mathcal{H}_t$	A one-dimensional row matrix of size $1 \times N$ at time t
h_i	An element of $\mathcal{H}_t$ at index i, then $h_i = 1$, if i is infected, 0, otherwise

- Adjacency List representation of contact types

$\mathcal{G}_{k,t}$	A list of M_t rows with size of each row equal to the number of contacts of type k at time t. Adjacency List can be stored with only size M_t because it can be stored as Hash Table, which allows arbitrary insertions and deletions at constant average cost per operation.
$g_{k,ij}$	An element of $\mathcal{G}_{k,t}$ in row i and column j, then $g_{k,ij} = b$, where $b \in \{1,\ 2, \dots, i-1, i+1, \dots, N\}$

- Properties of a contact network

$d_{k,i}$	A degree (i.e., number of contacts) for contact type $k \in \Omega$ for person i
$C_{k,i}$	A clustering coefficient (to represent transitivity) for contact type $k \in \Omega$ for person i
t_i	Number of triads that corresponds to person i
$P(d_k)$	Degree distribution for contact type $k \in \Omega$
$P(t\|d)$	Conditional distribution of number of triads given the degree

3.2 Algorithm

Overview: Only currently or previously infected persons and their immediate contacts (M_t number of people) are tracked individually as agent-based at time t. All other $N - M_t$ susceptible persons are modeled as a compartmental model. When contacts of infected persons become newly infected, their immediate contacts are generated using the algorithm below, such that, over time T, under the assumptions of a fully connected world and no recoveries or mortalities from infection, $M_t \to N$ as $t \to \infty$. Figure 3-1 shows simple illustration of ECNA.

Table 3-1: The Evolving Contact Network Algorithm

Step 1: Pre-assign degrees $\boldsymbol{d_{k,i}}$ for each node $\boldsymbol{i} \in \{\mathbf{1}, \mathbf{2}, \dots, \boldsymbol{N}\}$ from the distributions $\boldsymbol{P(d_k)}$

Step 2: Determine the initial infected contacts $\boldsymbol{n}$ in agent-based and update $\boldsymbol{\mathcal{H}_0}$

Step 3: Generate *close-contacts*: For each newly infected person $\boldsymbol{i}$, generate close contacts of each contact type $\boldsymbol{k} \in \boldsymbol{\Omega}$ by repeating the following steps.

1. Determine the number of new contacts of type, $\widehat{\boldsymbol{d}}_{\boldsymbol{k,i}}$, $\boldsymbol{k} \in \boldsymbol{\Omega}$ to generate
   i) If $\boldsymbol{C_{k,i}} = \mathbf{1}$ (e.g., family contacts), then $\widehat{\boldsymbol{d}}_{\boldsymbol{k,i}} = \boldsymbol{d_{k,i}} - \sum_{\boldsymbol{j}} \boldsymbol{a_{k,ij}}$
   ii) If $\boldsymbol{C_{k,i}} < \mathbf{1}$ and if $\sum_{\boldsymbol{j}} \boldsymbol{a_{k,ij}} < \boldsymbol{d_{k,i}} = \boldsymbol{F_D^{-1}(U[0,1])}$, then $\widehat{\boldsymbol{d}}_{\boldsymbol{i}} = \boldsymbol{d_i} - \sum_{\boldsymbol{j}} \boldsymbol{a_{k,ij}}$, else $\widehat{\boldsymbol{d}}_{\boldsymbol{i}} = \mathbf{0}$.
2. Generate $\widehat{\boldsymbol{d}}_{\boldsymbol{k,i}}$ contacts for a newly infected person (i.e., update $\boldsymbol{\mathcal{A}_{k,t}}$ or $\boldsymbol{\mathcal{G}_{k,t}}$ depending on the model):

2.1. Determine eligible persons to be a contact of $\boldsymbol{i}$:
Each of $\boldsymbol{M_t}$ persons in the agent-based and $\boldsymbol{N - M_t}$ persons in the compartmental, who satisfy Constraint 1 and Constraint 2 are eligible.

Constraint 1: Generating contacts that do not change the contact properties of previously infected persons, which can be determined as follows.

i) If $\boldsymbol{C_{k,i}} = \mathbf{1}$ (e.g., family contacts), $\boldsymbol{\beta} = \sum_{\boldsymbol{m}\in\boldsymbol{\Omega}}\{\boldsymbol{\mathcal{H}_t} + \boldsymbol{a_{m,i}}\} + \{\boldsymbol{\mathcal{H}_t}\boldsymbol{\mathcal{A}_k}\}$,
ii) If $\boldsymbol{C_{k,i}} < \mathbf{1}$, $\boldsymbol{\beta} = \sum_{\boldsymbol{m}\in\boldsymbol{\Omega}}\{\boldsymbol{\mathcal{H}_t} + \boldsymbol{a_{m,i}}\} + \{(\boldsymbol{\mathcal{H}_t} \circ \boldsymbol{a_{k,i}})\boldsymbol{\mathcal{A}_k}\}$, where $\circ$ is element-wise multiplication.

Then $\boldsymbol{\beta}$ will be a vector of size M, with $\boldsymbol{\beta_j} = \mathbf{0}$ if:

- $\boldsymbol{j}$ is not an infected contact (i.e., $\boldsymbol{\mathcal{H}}$'),
- not already a direct contact of $\boldsymbol{i}$ (i.e., $\boldsymbol{a_i}$), and
- not a contact of an infected contact of $\boldsymbol{i}$ (i.e., $(\boldsymbol{\mathcal{H}'} \circ \boldsymbol{a_i})\boldsymbol{\mathcal{A}}$ for $\boldsymbol{C_{k,i}} < \mathbf{1}$) to ensure maintenance of the clustering coefficient of $\boldsymbol{i}$, i.e., $\boldsymbol{c_i}$.

Therefore, all persons $\boldsymbol{j}$ with $\boldsymbol{\beta_j} = \mathbf{0}$ are eligible to form a contact with $\boldsymbol{i}$.

Constraint 2: Characteristics of the person match that randomly drawn from a probability distribution.

2.2. From among those eligible, choose $\widehat{\boldsymbol{d}}_{\boldsymbol{i}}$ persons at random.

To generate contact with one of the $\boldsymbol{M_t}$ persons in the agent-based, say $\boldsymbol{j}$, set $\boldsymbol{a_{k,ij}} = \mathbf{1}$ and with one of $\boldsymbol{N - M_t}$ persons in compartmental, first generate a new person in agent-based (increment $\boldsymbol{M_t}$ -transitioning them from compartmental).

3. Generate clustering: For each newly infected person $\boldsymbol{i}$, determine number of contacts to between uninfected contacts of $\boldsymbol{i}$.
   If $\boldsymbol{diag(\mathcal{A}U\mathcal{A})_i < t_i}$, then $\boldsymbol{\hat{t}_i = t_i - diag(\mathcal{A}U\mathcal{A})_i}$ where $\boldsymbol{U}$ is the upper triangular matrix of $\boldsymbol{\mathcal{A}}$, and $\boldsymbol{diag(\mathcal{A}U\mathcal{A})_i}$ is a number of existing triads for $\boldsymbol{i}$.
4. Generate $\boldsymbol{\hat{t}_i}$ number of edges (i.e., update $\boldsymbol{\mathcal{A}, \mathcal{G}}$) between contacts of $\boldsymbol{i}$. The contacts that are being contacted are randomly drawn and be satisfy Constraint 1.

Figure 3-2 illustrates the eligibility for newly infected contact.
Figure 3-3 illustrates the eligible edges between contacts of a newly infected.

Step 4: Determine transmissions from infected persons to immediate contacts
A susceptible person $\boldsymbol{i}$ of the $\boldsymbol{M_t}$ has an infection risk of $\boldsymbol{\theta = 1 - (1 - p)^k} \quad \forall \boldsymbol{i}$, where $\boldsymbol{p}$ is the disease transmission risk, and $\boldsymbol{k}$ is the total number of infected contacts.

Step 5:
Update the time step and Go to **Step 3**.

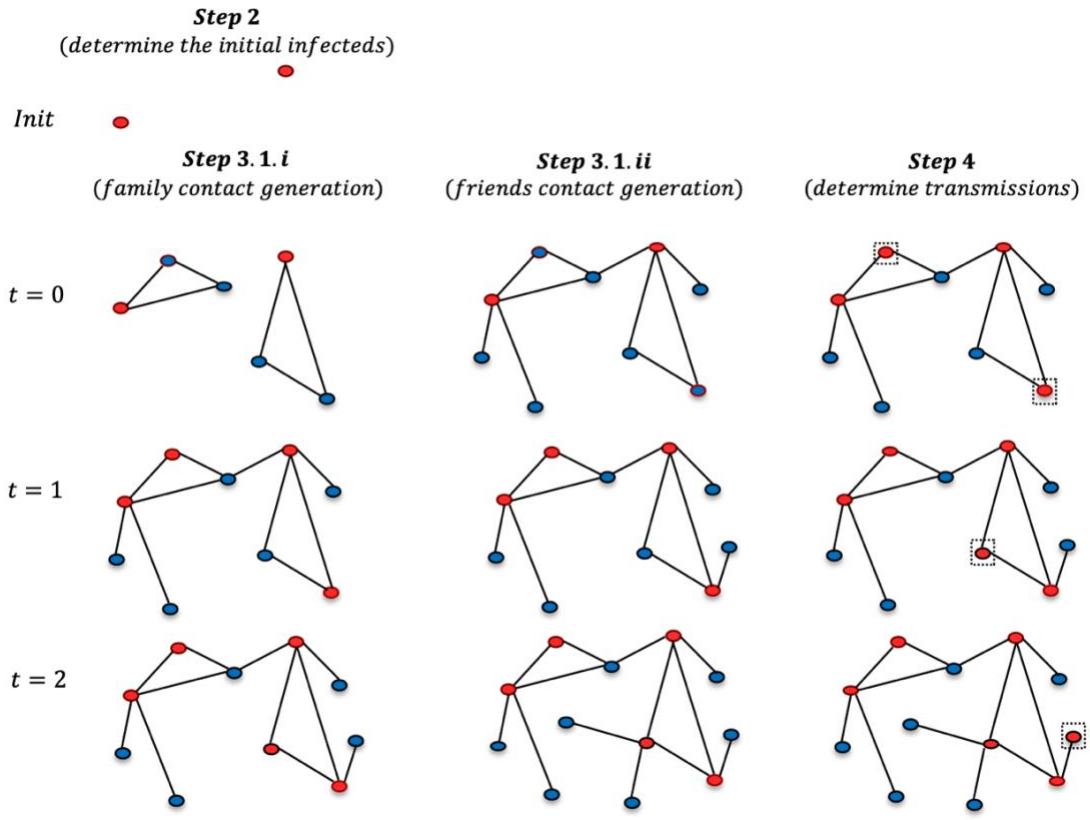


Figure 3-1: A schematic representation of the ECNA with $N = 12, n = 2, k = 2$.

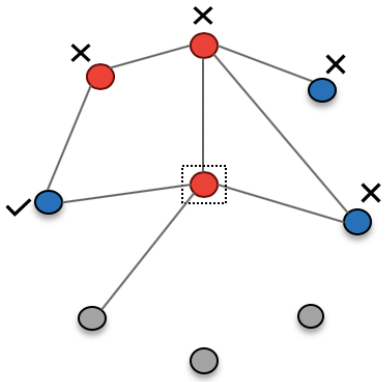


Figure 3-2: Eligible contacts for newly infected

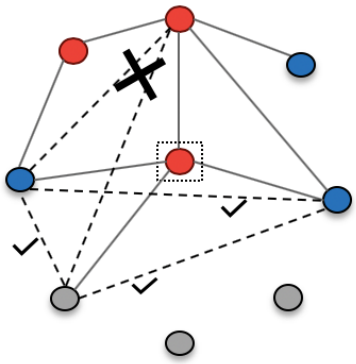


Figure 3-3: Eligible edges for newly infected

In the figures, the infected, uninfected and people who are in the compartmental model are colored in red, blue and gray, respectively. The checkmark indicates it is eligible, and the cross mark indicates it is ineligible.

3.3 Empirical Validation of the Algorithm

As our method is attempting to replace ABNM, we use ABNM as a benchmark and validate our model by comparing its results with that generated by ABNM. Specifically, we compare the following metrics which are key parameters for epidemic prediction:

i. The number of infections over time: which is a proxy for epidemic predictions

ii. Average degree: The average number of contacts in the network should match population data. In ABNM, this is an input. In ECNA, this is an outcome because people are added when their contacts become infected. Thus, as the network grows and the full population becomes infected, we would expect that the average degree will match that of ABNM.

The general validation process is illustrated in Figure 3-4, and consists of the following steps:

a) Construct ABNM

b) Collect empirical network data from ABNM to generate ECNA. Note that, typically, network data would be taken from the population under study. However, for this thesis, for purposes of testing only, we generate hypothetical data using the ABNM.

c) Construct ECNA

d) Extract validation parameters (number of infections, and average degree) from ECNA and ABNM.

e) Compare ECNA results with ABNM

We compared multiple types of graphs as discussed in the next section.

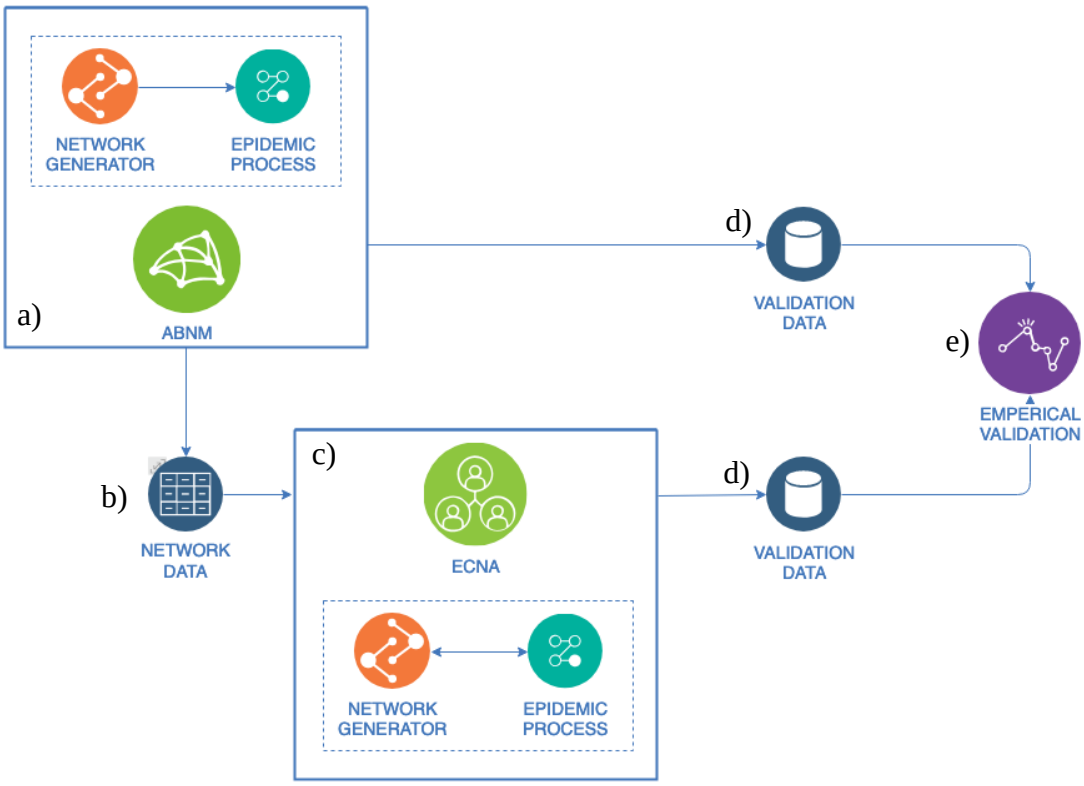


Figure 3-4: Validation pipeline of the ECNA. It first collects data, which varies depending on the model, from ABNM then generate the evolving network while simulating the epidemic using the ECNA and compare the results with results from ABNM.

3.4 Baseline ABNM Model

As discussed above, in the application of ECNA, network data needed as inputs to the model would be collected from the population under study. However, for this thesis, for purposes of testing only, we generate hypothetical empirical data by generating a simple ABNM. Specifically, we collect data related to degree distribution.

3.5 Models

The algorithm in Section 3.2 is written using an Adjacency Matrix representation of a graph, particularly in finding eligible contacts from a population M_t. This section introduces different implementations of a graph using a different graph representation along with ECNA adjustments to each model. Such graphs can be denoted by $G(V, E)$, where V is a set of vertices and E is a set of edges.

The models described below were implemented in MATLAB, Java, and Python.

Model 1.0 focuses on a proof of concept of ECNA, by implementing it on a network with open degree distribution that has multiple contact layers. This uses Adjacency Matrix graph representation which is less error-prone because of its numerical accuracy when computing eligible contacts (Step 3 of the Algorithm). Configuration model was used for developing the ABNM for comparison.

Model 1.1 also implements ECNA on a network with open degree distribution that has multiple contact layers. However, it focuses on the computational efficiency of the algorithm by implementing it on the same network structure as we used on Model 1.0 but using *hashable* Adjacency List graph object in Java instead of adjacency matrix.

Model 2.0 focuses on the flexibility of the algorithm by implementing it on different network type, scale-free networks, and applies it to a multi-community setting. It uses the

NetworkX package in Python. Preferential attachment algorithm was used, for generating a scale-free network, in an ABNM for comparison.

3.5.1 Model 1.0: Multi-contact Evolving Network using Adjacency Matrix

This model provides an implementation of ECNA on a multi-contact network using an *adjacency matrix representation* of a graph. The network has two different contact structures: i) The nodes are grouped into fully connected, i.e., clustering coefficient is equal to 1, ii) The nodes are grouped with some clustering, i.e., clustering coefficient is less than 1 but more than 0.

Figure 3-5 shows a structure of a multi-contact network, where family contact (inside circles) is fully connected whereas friends (outside circles) is not. Figure 3-6 illustrates that each contact type network can be represented as a network layer which allows representing each network type with adjacency matrix $\mathcal{A}$.

The friends contact network follow degree distribution $P(d)$ and has clustering coefficient $C_{k,i} < 1$. For simplicity purposes, the household size is assumed to be a fixed number, 3 in this model, for each house. However, the household size can follow a degree distribution in a more significant expansion of a model.

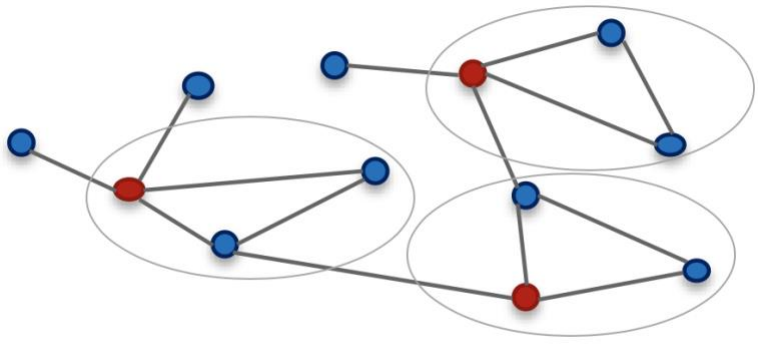


Figure 3-5: Multi-contact network

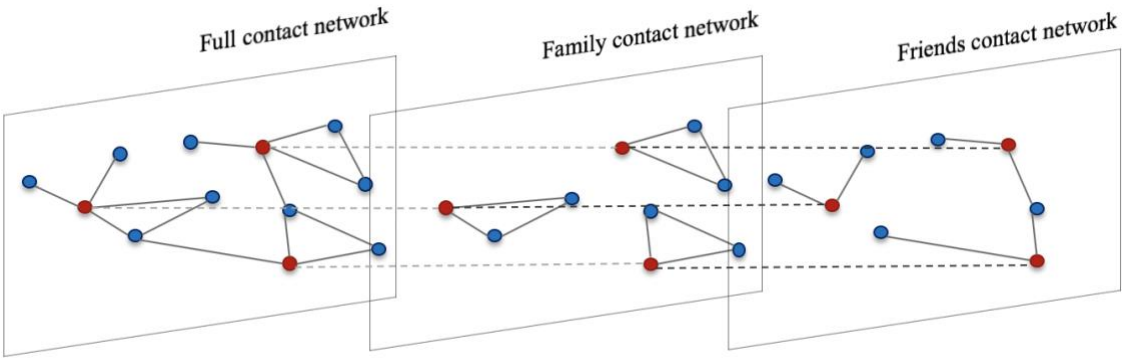


Figure 3-6: Contact types describe the different types of interactions among agents. The dashed lines emphasize that the graphs have the same nodes, but the edges are distinct.

3.5.1.1 Base ABNM

There exist many different tools to assist in the development of agent-based models. Among these models, NetLogo was used to develop the baseline ABNM model because NetLogo provides a graphical tool for quickly constructing interfaces as well as it is highly recommended for simple models (M. Berryman 2008; M. J. Berryman and Angus 2010).

Therefore, we developed the baseline ABNM for Model 1.0 using NetLogo. The NetLogo world is built up of agents that can follow instructions. In our model, a *turtle* agent represents a person, and a *link* agent visually serves as a line connecting two *turtles*.

The ABNM in NetLogo enables us to enter the network and epidemic properties as inputs to the model and provides a constructed network visualization with infection dynamics graphics. With given network properties of a number of houses, average household size, and average clustering coefficient of friends, it randomly links turtles together until it matches its input values.

Also, more information such as degree distribution can be printed on the command center as well as conveniently stored into CSV files for collecting data for the ECNA

models. Figure 3-7 shows a NetLogo graphical user interface (GUI) for the ABNM model for constructing a network with two types of contacts of family and friends, then simulating the spread of infection over.

Though NetLogo provides an interactive GUI and relatively convenient to use and learn, its lack of right object-oriented features could make some things difficult, and NetLogo is often slow to compare to Java-based platforms such as MASON and Repast (M. Berryman 2008).

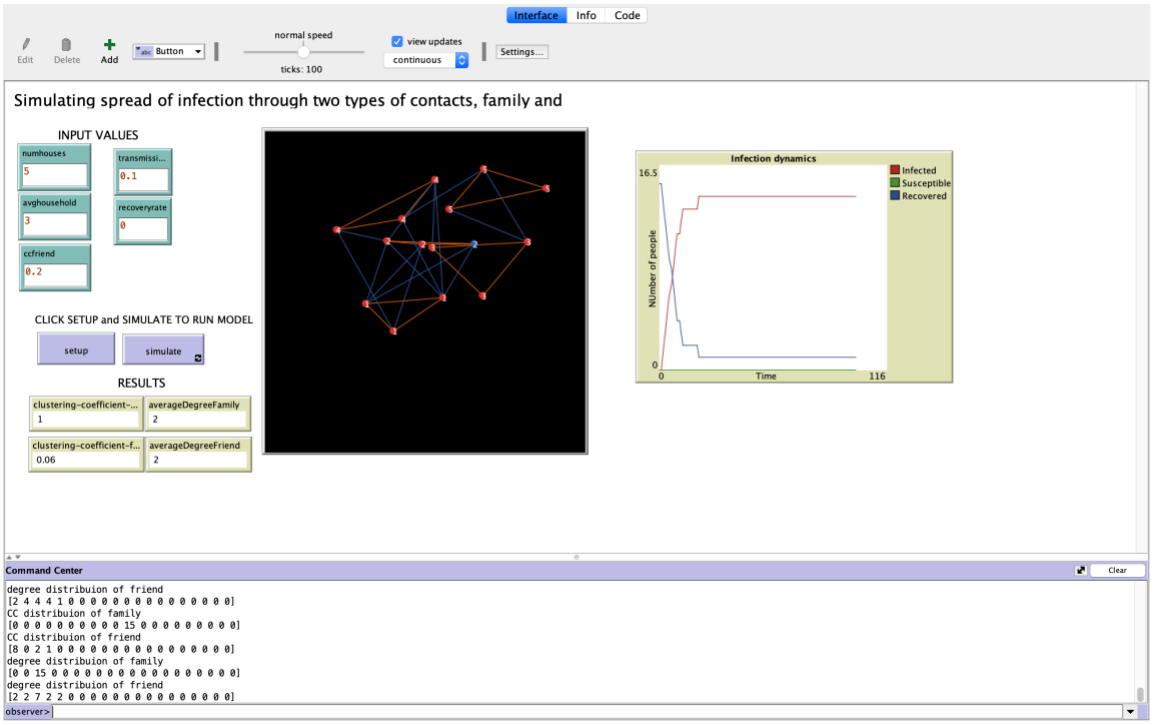


Figure 3-7: NetLogo graphical interface, where a number of input values can be entered, for an ABNM. Red lines represent family contacts, and blue lines represent friend contacts.

3.5.1.2 Objective of the model

The objective of Model 1.0 for a multi-contact network structure generator is testing the accuracy of the ECNA by comparing disease incidence on a network that is generated

using the ECNA with the disease incidence on a network that is generated using ABNM given the same network.

3.5.1.3 Model parameters

The network properties can be controlled by a user using the model parameters summarized in Table 3-2.

Table 3-2: Model 1.0 Parameters

Parameter	Description
$\boldsymbol{N} \in \mathbb{N}^{+}$	Number of nodes (population)
$\boldsymbol{n} \in \mathbb{N}^{+}$	Number of initially infected people
$\lvert\boldsymbol{k}\rvert \in \mathbb{N}^{+}$	The number of contact network types
$\boldsymbol{P}(\boldsymbol{d_k})$	Degree distributions of the contact networks
$\boldsymbol{C_k}$	Clustering coefficients of the contact networks
$\boldsymbol{p_{prev}} \in [\mathbf{0}, \mathbf{1}]$	The percentage of prevalence
$\boldsymbol{\beta} \in [\mathbf{0}, \mathbf{1}]$	The disease transmission rate

3.5.1.4 The data structure, complexity, and technology

Model 1.0 uses *an adjacency matrix representation* of a graph, which is described in 2.5 and aims to do an empirical analysis of ECNA on its both epidemiological and network properties by doing a simulation on a small population. This model was developed in MATLAB because MATLAB is designed to operate primarily on whole matrices and arrays ("Matrices and Arrays - MATLAB & Simulink" n.d.).

The algorithm for Model 1.0 is divided into two parts:

i. Initializations of the graphs using adjacency matrices and infection state matrix (lines 1-2 in Table 4-3). Here it initializes the zero matrices with a dimension of $\boldsymbol{N} \times \boldsymbol{N}$ because it is best to preallocate space for the largest matrix

that is anticipated to be created using MATLAB ("Creating, Concatenating, and Expanding Matrices - MATLAB & Simulink" n.d.). Then, update the elements of the matrix when the graph grows from size n to N.

Although starting with $\boldsymbol{N \times N}$ sparse matrix is both space and time efficient than starting with $n \times n$ then expanding the matrix whenever the graph grows, the cost of sparse matrix multiplication is expensive. Model 1.1 was formulated to solve this computational challenge, and Section 3.5.2 provides detailed information on the model.

Table 3-3: Model 1.0 Algorithm

Algorithm	
1. Initialize $\boldsymbol{N \times N}$ zero matrices $\boldsymbol{\mathcal{A}_k,\ k \in \Omega}$	
2. Initialize $1\times \boldsymbol{N}$ zero matrix $\boldsymbol{\mathcal{H}}$	
3. Follow algorithm in 3.2	

3.5.2 Model 1.1: Multi-contact Evolving Network using Adjacency List

Model 1.1 provides an implementation of ECNA on a multi-contact. i.e., family and friends, network using an *adjacency list representation* of a graph. While Model 1.0 focuses on a numerical validation of infections by directly implementing the algorithm, Model 1.1 focuses on a computational efficiency of the algorithm by implementing the algorithm with an efficient data structure.

3.5.2.1 Base ABNM

The base ABNM is the same model that is used for Model 1.0 using NetLogo.

3.5.2.2 Objective of the model

The objective of Model 1.1 for a multi-contact network structure generator is testing the computational efficiency of the ECNA by comparing the computation time of ECNA that used the adjacency list of a graph for a single run with the computation time of ABNM given the same network for a single run.

3.5.2.3 Model parameters

The same as Model 1.0 in Section 3.5.1.2.

3.5.2.4 The data structure, complexity, and technology

Model 1.1 uses *an adjacency list representation* of a graph, which is described in 2.5.2 and aims to do an empirical analysis of ECNA on its epidemiological properties as well as computational efficiency by doing a simulation on a larger population. This model was developed in Java using $HashSet$ class in JAVA because of its constant time operations as shown in Table 3-4. Because of Object-Oriented Programming and HashSet representation, the formulation of the algorithm and model modified as shown in Table 3-5. Finding eligible contacts using the Adjacency List has to be changed from matrix multiplication form, and the pseudocode is in Table 3-6.

Table 3-4: Java HashSet complexity

Java Collection	Add	Remove	Contains	Size	Data Structure
$HashSet$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$HashTable$

Table 3-5: ECNA algorithm using Adjacency List

Algorithm 1	
1. Initialize empty adjacency list $\mathcal{G}_k,\ k \in \Omega$	
2. Initialize empty sets of integers for $\boldsymbol{newlyInfected}$ and $\boldsymbol{oldInfected}$	

3. Follow algorithm in 3.2 with following:
 	
Step 2:
Determine the initial infected contacts n in agent-based and update $\boldsymbol{newlyInfected}$
Step 3: for i in $\boldsymbol{newlyInfected}$:
for k in Ω:
$\hat{d}_{k,i} = d_{k,i} - \mathcal{G}_{k,i}.size()$
for j in $\hat{d}_{k,i}$:
$\boldsymbol{eligibles}$ = find $\boldsymbol{eligible}$ people from *the agent-based population* (**Algorithm 2**)
$\mathcal{G}_{k,i}$ += a neighbor from the $\boldsymbol{eligibles}$, if any
$\mathcal{G}_{k,i}$ += a neighbor from the *compartmental* population
for j in $\mathcal{G}_{k,i}$:
generate clustering
update $\mathcal{G}_{k,i}$ accordingly
Step 4: Determine transmissions
Step 5: Update time step and go to Step 3	

Table 3-6: Finding eligible contacts using Adjacency List

Algorithm 2	
for jj in $\mathcal{G}_k.keys().size()$: if $isEligible(jj, k, i)$ **update** $\boldsymbol{eligibles}$	
$isEligible(jj, k, i)$: *if ($jj == i$ or jj is infected)* return false if $\mathcal{G}_{k,jj}.size() >= (d_{k,jj})$: return false if jj is a contact of an infected contact of i return false if jj and i are already contacted return false return true	

3.5.3 Model 2.0: Multi-communities Evolving Network using *NetworkX* and Preferential Attachment

The spread of infectious diseases that transmit through *close-contact* such as Ebola or HIV depends on heterogenous mixing within the population. This mixing takes numerous individual information such as population size and density (Suryaprasad et al. 2013), the age structure of the population (Merli and Hertog 2010), the composition of household (Adams 2016; Cauchemez et al. 2009), and demographic and cultural practices (Alexander et al. 2015) into account. For example, the vast majority of HIV transmissions, approximately 50,000 transmissions per year from 2007 through 2010 in the U.S., were from sexual contact (Eubank et al. 2004). CDC classified the HIV transmission category as male-to-male sexual contact that includes both homosexual and bisexual contact, and heterosexual female contact (CDC, 2012). Therefore, it is important to incorporate different sexual behaviors and their mixing between the groups when simulating a sexually transmitted disease. Thus, Model 2.0 focuses on applying the ECNA to generate a network with two sexual contact groups and testing its flexibility on a multi-community structured network when there is mixing between the communities.

It is observed that the network of human sexual contacts is scale-free, that is, the distribution follows a power-law with an exponent between 2 and 3 (Liljeros et al. 2001; Barabási, Ravasz, and Vicsek 2001; Schneeberger et al. 2004). Scale-free networks can be formed using a preferential-attachment mechanism (Barabási, Ravasz, and Vicsek 2001).

However, preferential-attachment mechanism cannot be directly used for the ECNA network generation since it attaches nodes to a node with probability that is proportional to its current degree whereas the ECNA requires to know degree of newly infected node and its degrees of neighbors as soon as it becomes infected (Eden et al. 2018).

Therefore, the previous study (Eden et al. 2018) presented a model that fits a non-linear neural network model to data from multiple scale-free networks and integrated to ECNA to generate a scale-free network.

Previous work focused on a single community and trained a neural network on a scale-free network that has a single community. In this thesis, in Model 2.0 we applied ECNA to a two-community network.

3.5.3.1 Base ABNM

To test Model 2.0, we need a baseline ABNM model that simulates the infection on a network with two sexual contact groups that has some mixing between communities.

Therefore, we built a model that generates this hypothetical network with two sexual contact groups, each following a power-law, and with mixing between the two groups. As an example, the two groups can represent 'heterosexual men and women' and 'gay men', and the mixing represents 'bisexual men', a categorization typically used for HIV modeling. To develop the base ABNM model, we first generated two graphs, $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$, using preferential attachment and rewired edges while keeping the degree distribution the same to incorporate mixing between two communities. Figure 4-8 illustrates the rewiring process which follows the algorithm below:

i. Pick an edge e_k between nodes $(v_{1,i}, v_{1,j})$ that has degrees $(d_{1,i}, d_{1,j})$ where $e_k \in E_1$ and $v_{1,i}, v_{1,j} \in V_1$

ii. Pick an edge e_l between nodes $(v_{2,i}, v_{2,j})$ that has degrees $(d_{2,i}, d_{2,j}) = (d_{1,j}, d_{1,i})$ where $e_l \in E_2$ and $v_{2,i}, v_{2,j} \in V_2$

iii. Remove e_k and e_l from $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$,respectively

iv. Create edges between nodes $(v_{1,i}, v_{2,j})$ and $(v_{2,i}, v_{1,j})$

We built this preferential attachment of communities network generator using *NetworkX* package and its in-built preferential attachment graph generator function in Python. Figure 4-9 shows an example resulting network of $|V_1|=|V_1| = 20$ and 10 percent mixing after the rewiring process. Figure 4-10 shows a pictorial proof of degree distributions of $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ stays the same after rewiring and resulting graph G is scale-free.

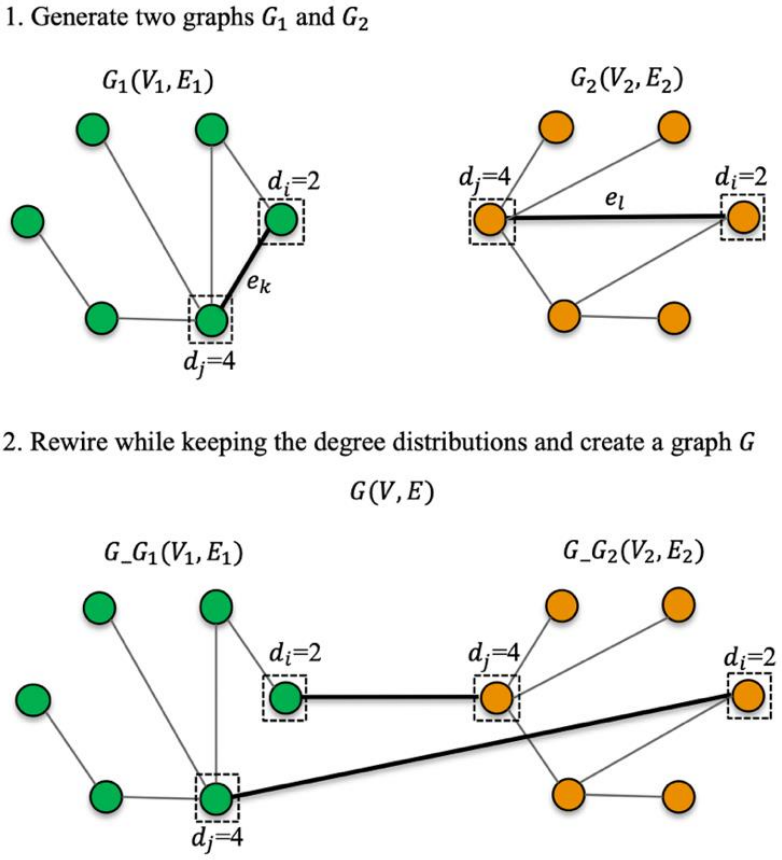


Figure 3-8: Generating a network with two groups where each group is scale-free from two independent scale-free networks by removing edges from each network then adding edges between networks.

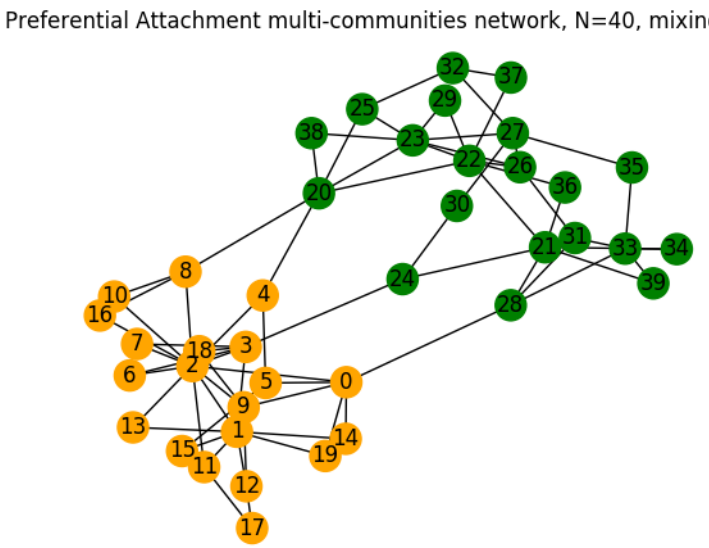


Figure 3-9: A final network, the baseline network for ANBM, with two groups where each group is scale-free.

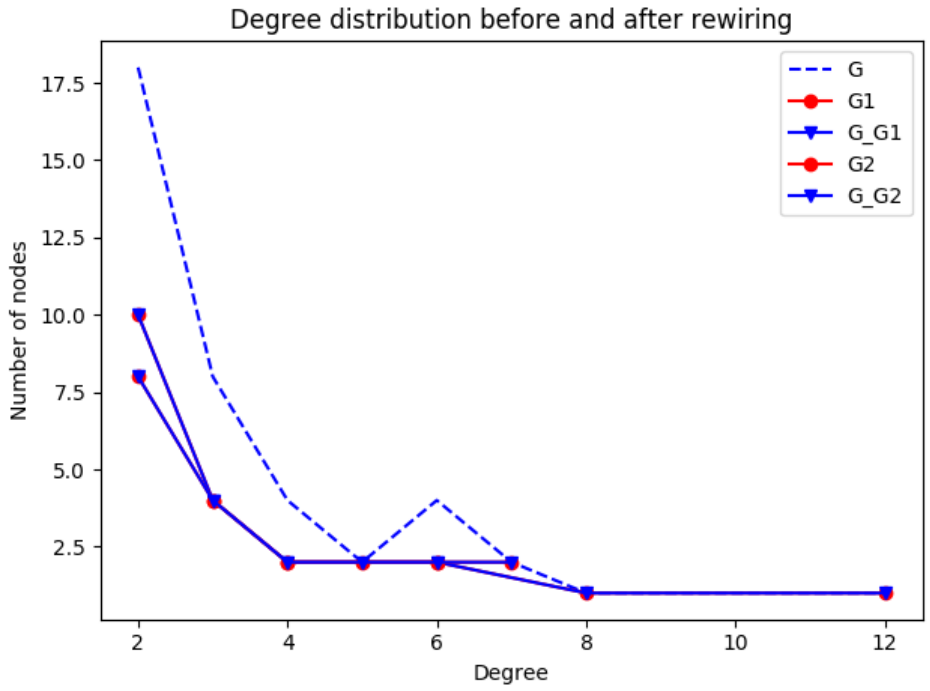


Figure 3-10: Degree distributions of the generated network in Figure 3-9 before and after rewiring the edges

3.5.3.2 Objective of the model

The objective of Model 2.0, a multi-community network generator using ECNA, is testing the flexibility of the ECNA by applying to a network with two-communities that has a mixing between communities and by integrating the trained Neural Network model.

3.5.3.3 Model parameters

Table 3-7: Model 2.0 Parameters

Parameter	Description
$\boldsymbol{N} \in \mathbb{N}^{+}$	Number of nodes (population)
$\boldsymbol{n} \in \mathbb{N}^{+}$	Number of initially infected people
$\lvert\boldsymbol{k}\rvert \in \mathbb{N}^{+}$	The number of communities
$\boldsymbol{\mu} \in [\mathbf{0}, \mathbf{1}]$	The mixing between communities
$\boldsymbol{P}(\boldsymbol{d_k})$	Degree distributions of the contact networks
$\boldsymbol{\mathcal{M}_{k,b}} \in [\mathbf{0}, \mathbf{1}]$	Mixing probability of each community
$\boldsymbol{p_{prev}} \in [\mathbf{0}, \mathbf{1}]$	The percentage of prevalence
$\boldsymbol{\beta} \in [\mathbf{0}, \mathbf{1}]$	The disease transmission rate

3.5.3.4 The data structure, complexity, and technology

Model 2.0 aims to do empirical analysis on the flexibility of ECNA by implementing ECNA to a multi-community network along with the neural network model that predicts the cumulative distribution of degrees of neighbors of a newly infected person.

This model was developed in Python using the *NetworkX* package which was primarily designed for general network analysis as well as a platform for developing new algorithm and theory (Hagberg, Swart, and S Chult 2008). This package provides many different types of network generators and graph objects that represent both undirected graphs, directed graphs and more. The nodes in *NetworkX* graph is *hashable* Python object;

therefore, it conveniently provides many functions such as getting degree and clustering coefficient of a node.

Model 2.0 focuses on generating a two-community network with mixing in between and integrating the neural network model; the general algorithm of this model is modified accordingly.

Table 3-8: Model 2.0 Algorithm

Step 1: i. Determine the initial infected contacts $\boldsymbol{n/2}$ in each community so that transmission process on both group and update $\boldsymbol{\mathcal{H}_0}$. ii. Determine the degrees of newly infected contacts and the degrees of their neighbors
Step 2: Generate *close-contacts*: For each newly infected person $\boldsymbol{i}$, generate close contacts of each community type $\boldsymbol{k} \in \boldsymbol{\Omega}$ by repeating the following steps. 1. Determine the number of new contacts of type, $\widehat{\boldsymbol{d}}_{\boldsymbol{k,i}}$, $\boldsymbol{k} \in \boldsymbol{\Omega}$ to generate by subtracting current degree from the prescribed degree 2. Generate $\widehat{\boldsymbol{d}}_{\boldsymbol{k,i}}$ contacts for a newly infected person from either eligible contacts or undiscovered 3. For each $\widehat{\boldsymbol{d}}_{\boldsymbol{k,i}}$ contacts of newly infected, find the distributions of degree of neighbors 4. Assign degrees to neighbors of $\widehat{\boldsymbol{d}}_{\boldsymbol{k,i}}$ contacts
Step 3: Determine transmissions from infected persons to immediate contacts A susceptible person $\boldsymbol{i}$ of the $\boldsymbol{M_t}$ has an infection risk of $\boldsymbol{\theta = 1 - (1 - p)^k}$ $\forall \boldsymbol{i}$, where $\boldsymbol{p}$ is the disease transmission risk, and $\boldsymbol{k}$ is the total number of infected contacts.
Step 4: Update the time step and Go to step 3.

CHAPTER 4

4 RESULTS

Our aim was for the ECNA to produce approximately similar results as an ABNM, i.e., both epidemiological and network properties converge to that of ABNM as contact network evolves into the full population. Simple hypothetical networks were generated for each model to test the convergences, and the results after the epidemic process simulated by ECNA and ABNM were compared in each model.

Model 1.0 tested a proof of concept of the ECNA by implementing it using an Adjacency Matrix representation of a graph on small size networks with possible parameters.

Model 1.1 tested the computational efficiency of the ECNA by implementing it using an Adjacency List representation of graph on larger size networks with possible parameters.

Model 2.0 tested the flexibility of the ECNA by applying it to a multi-community structured network with possible parameters.

4.1 Model 1.0

To test a proof of concept of the ECNA in Model 1.0, we simulated two types of contacts, which have high clustering, in a population of 100 persons. Then, simulated the spread of possible diseases using the ECNA and ABNM under different network properties of degree, d, and clustering coefficient, cc, for family and friends contacts and different transmission probabilities, p, to represent different diseases. We compared epidemic projections on six different scenarios that are a combination of different network properties

and transmission probability parameters (Table 4 1). We run 100 simulations using both ECNA, ABNM, and a compartmental model to compare the epidemic projections in each scenario.

We assumed the household size is equal to 3 and the average degree of friends were based on a study by Read, Eames, and Edmunds (Read, Eames, and Edmunds 2008). They conducted a diary-based survey to study dynamic social network and infectious disease spread on the network and found the daily encounters of people (i) all contact types (mean, 14. 29), (ii) contacts that were conversational only (mean, 12.3) and (iii) contacts that included skin-to-skin physical contact (mean, 1.99). The ECNA is for diseases that transmit through close-contact; therefore, we used the mean degree of 1.99 of skin-to-skin physical contact information from the study and set the average degrees of friends contact network to 2 and 4.

Table 4-1: Network properties of different networks that are used in Model 1.0

	Population (N)	Initial number of infected persons (n)	Household (family) size, cc=1	Average degree (d) of friends contact	Average clustering coefficient (cc) of friends contact	Transmission probability (p)
1	100	1	3	2	0	0.1
2	100	1	3	2	0	0.5
3	100	1	3	2	0.2	0.1
4	100	1	3	2	0.2	0.5
5	100	1	3	4	0.4	0.1
6	100	1	3	4	0.4	0.5

4.1.1 Epidemic projections and network properties

To test the convergence of the ECNA, we compared the results from 100 simulations of a hypothetical population of 100 persons using the ECNA and ABNM under six different scenarios. Shown in individual chart title are d and cc for friends contacts since family contacts properties are the same in each scenario.

A deterministic compartmental (population level) model, red dashed line in Figure 4-1, was also used for simulating infectious disease spread for a demonstration of difference with agent-based models (ABNM and ECNA).

4.1.1.1 Prevalence

We have compared the total number of infections at each simulation time step in ABNM, ECNA and compartmental models after 100 simulations and their 5^{th}, 50^{th} and 95^{th} percentiles. Figure 4-1 shows those epidemic projections by the ECNA result similar to ABNM in each scenario. In contrast, the compartmental model overestimates the infection cases, and the average is at about the 95th percentile of agent-based models (ECNA, ABNM) due to the assumption of random mixing.

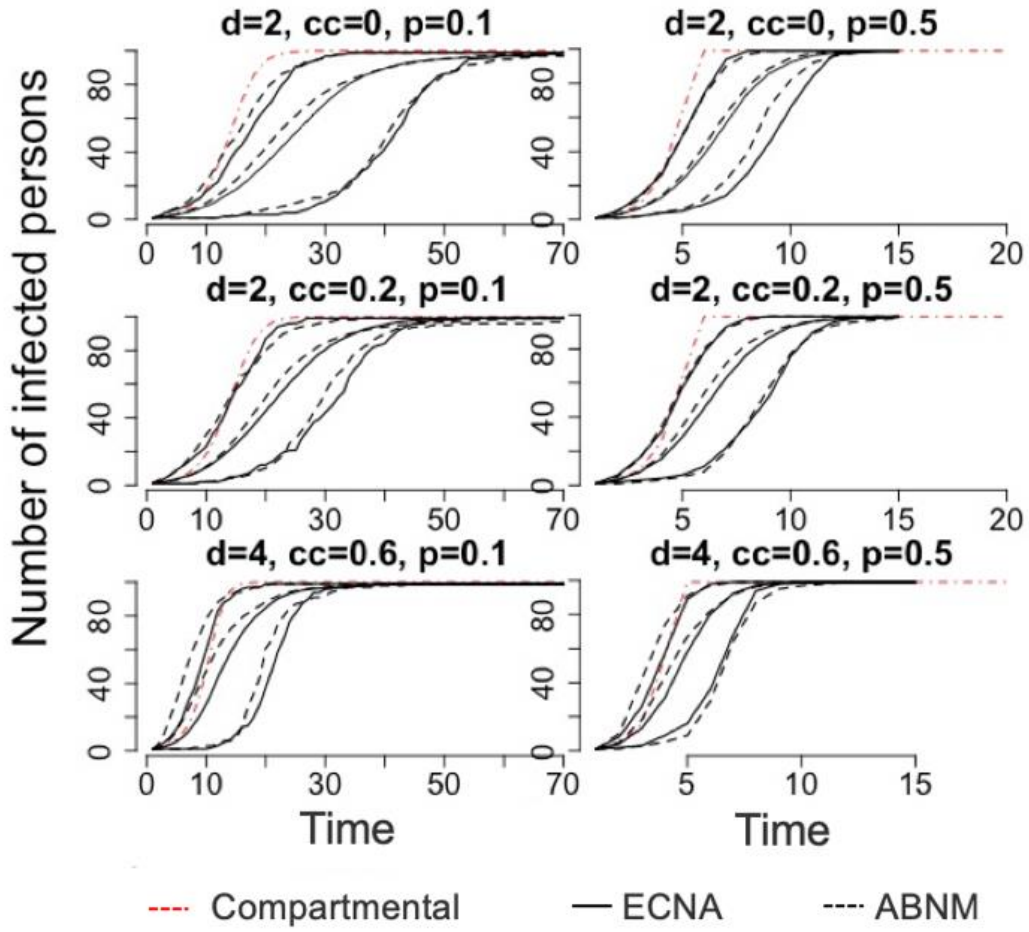


Figure 4-1: Epidemic projection comparison of the ECNA (Model 1.0), ABNM and compartmental model. 5th, 50th, and 95th percentiles of 100 simulations.

4.1.1.2 Incidence

We compared the number of newly infected people, at each simulation time step in ABNM, ECNA and compartmental models of 100 simulations and their 5^{th} and 95^{th} percentiles. Figure 4-2 shows (example two scenarios of the six scenarios) that the trend of newly infected is similar in ABNM and ECNA, while the compartmental model reaches its peak earlier.

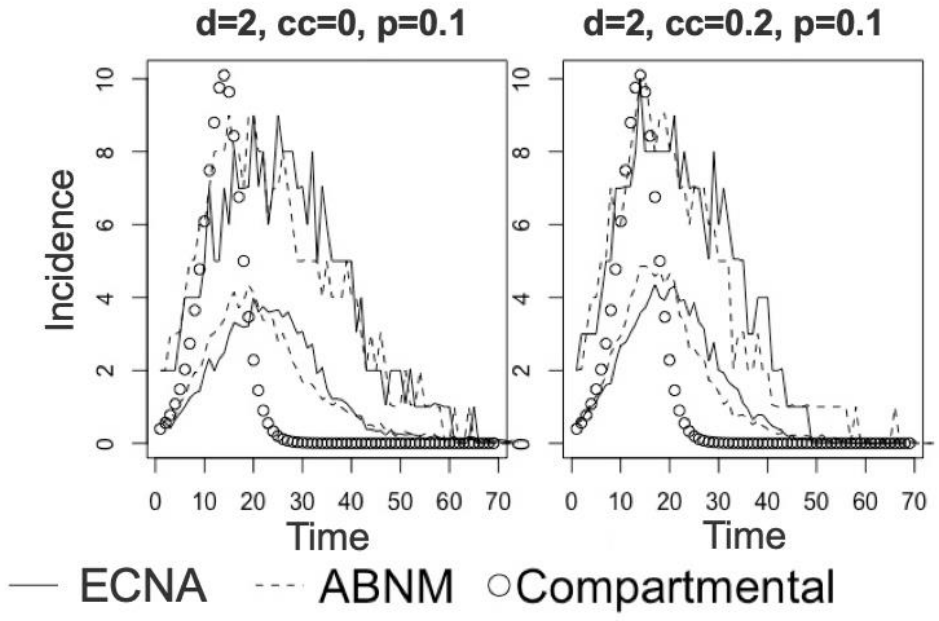


Figure 4-2: Incidence projection comparison of the ECNA (Model 1.0) and ABNM. 5th and 95th percentiles of 100 simulations.

4.1.1.3 Network Properties

Figure 4-3 shows the comparison of network properties, average degrees of infected friends contact, at each simulation time step in the ECNA model of 100 simulations and their 5th, 50th and 95th percentiles. The ECNA generates only infected people and their contacts while simulating the disease transmission while maintaining the network properties; the results show that the average degrees of infected people converge to population averages in ABNM and the trend shows that people with more contacts have higher chances of getting infected as expected.

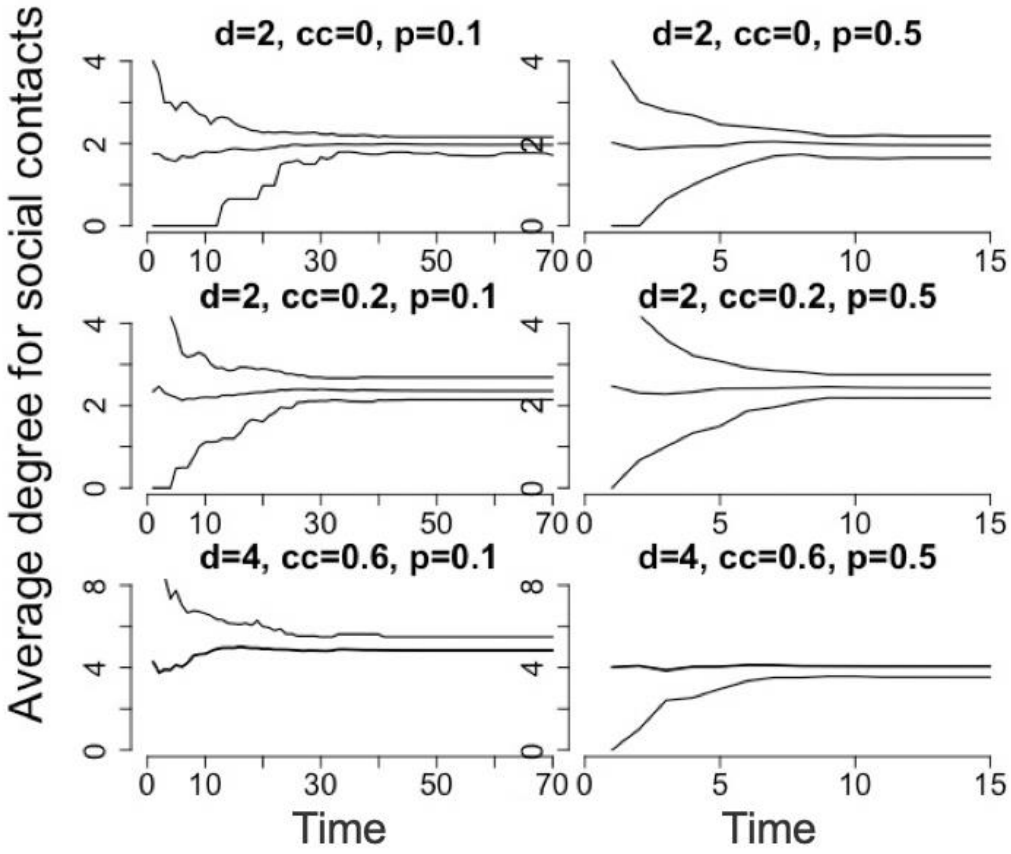


Figure 4-3: Model 1.0 the convergence of average degrees of friend's contact

4.1.1.4 Programming debugging

For MATLAB code error diagnosis purposes, we compared the simulated new infections, s_t, and the expected new infections, e_t, at time t using the data that was generated using the MATLAB code. Figure 4-4 shows the comparison between simulated new infections and the expected new infections using the following equations on the network size of 1500 with p=0.1 and overlapping lines suggest that MATLAB code works as expected.

$s_t = inf_t - inf_{t-1}$, where inf_t is the number of infected persons at time t.

$e_t = (1 - (1 - \beta)^{c_i})D_t$, where β is the transmission rate, c_i is the average number of contacts of uninfected person i at time t, and D_t is the number of uninfected persons in the network (people who are discovered/generated)

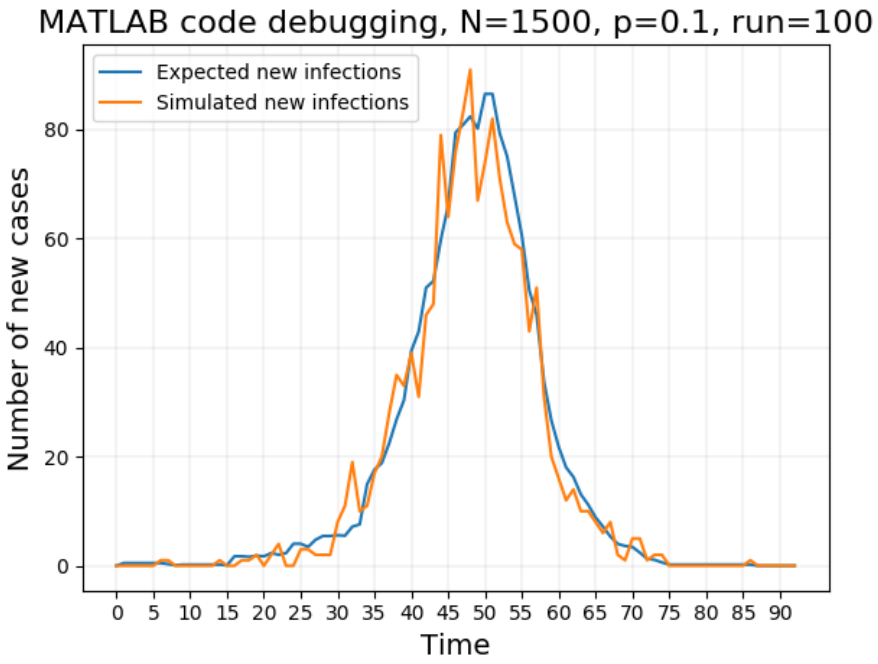


Figure 4-4: Program debugging in MATLAB code

4.1.2 Computation Time

The second hypothesis of Model 1.0 was that this model, which used an adjacency matrix representation of a graph, is computationally more expensive than ABNM due to large matrix operations when generating and selecting eligible contacts to add to the network. We compared the computation time of a single simulation on different networks size of 400, 600, 900 and 1500 with transmission probability $p = 0.01$ using both Model 1.0 (MATLAB) and ABNM (NetLogo) on a standard desktop. Figure 4-5 shows that computation time of ECNA on MATLAB increases faster than ABNM on NetLogo when the population size increases as expected.

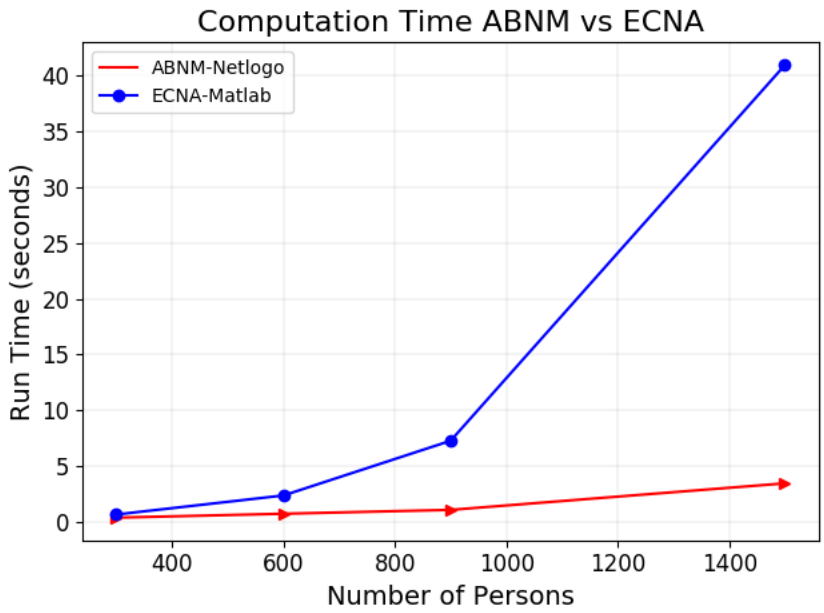


Figure 4-5: Computation time comparison of ECNA (Model 1.0) and ABNM

4.1.3 Graph Visualizations

An *animation* technique, one of the validation technique, provides model results graphically during the simulation run. Figure 4-6 shows how the network is being evolved using the ECNA and it was plotted using MATLAB. The thicker blue line represents family contacts while thinner blue line represents friends contacts and red node indicates infected while blue node indicates susceptible person.

These graphs show that Model 1.0 successfully generates a network with two contact types, family and friends, while simultaneously simulating the transmission of infection over the population using the ECNA.

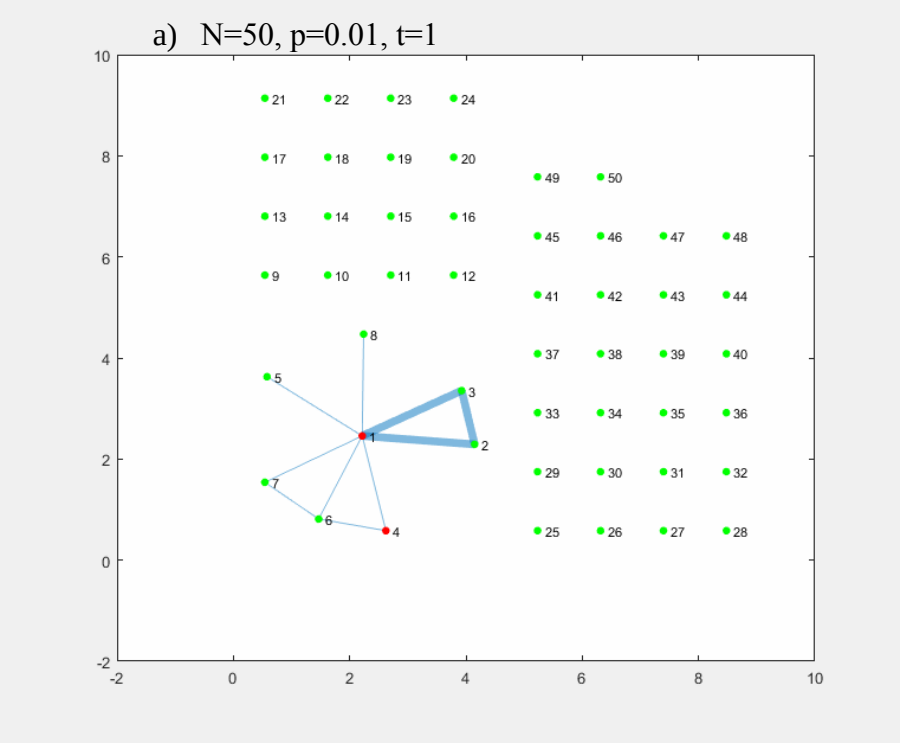


Figure 4-6: An evolution of a network with two-contact type using the Evolving Contact Network Algorithm. a) An infected contact network at the initial stage of the outbreak b) An infected contact network where the epidemic spread over.

4.2 Model 1.1

The purpose of Model 1.1 is to test the computational efficiency of the ECNA in using the same algorithm as Model 1.0 but developed in Object-Oriented Programming in Java to implement an Adjacency List graph object. Therefore, we compared the computation time of a single simulation on different networks sizes using both Model 1.1 (Java) and ABNM (NetLogo) on the same standard desktop. Also, we included the epidemic projections comparison results on a network size of 6000 with a transmission rate of 0.01, which was based on HIV transmission risk data (Patel et al. 2014), as a validation of the Java implementation.

We have included computation time result based on 20 different sizes of contact network. The hypothetical contact networks that are used in this model all have the same network properties except the network size of a range of 300 to 150,000. All models start with one infected node, and the network properties of average degree and clustering coefficients are 2 and 1 for family contact type and 2.5 and 0.2 for friends contact type, and the transmission probability is 0.01 in Model 1.1.

4.2.1 Epidemic projection and network property

4.2.1.1 Prevalence

Figure 4-7 shows a result from 100 simulations of a hypothetical population of 6000 persons using the ECNA, ABNM and compartmental model and their 5th, 50th and 95th percentiles of the total number of infections at time step 200. It shows those epidemic projections by the ECNA result similar to ABNM. Like Model 1.0, a compartmental model

overestimates the infection cases, and the average is at about the 95th percentile of agent-based models (ECNA, ABNM) as expected.

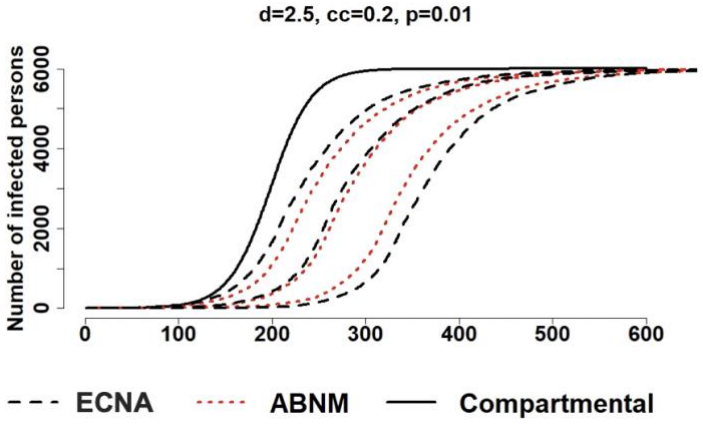


Figure 4-7: Epidemic projection comparison of the ECNA (Model 1.1), ABNM and compartmental model. 5th, 50th, and 95th percentiles of 100 simulations.

4.2.1.2 Network Properties

Figure 4-8 shows the comparison of network properties of 100 simulations on a network of 6000 population and their 5^{th}, 50^{th} and 95^{th} percentiles. It shows that the average degrees of infected people converge to population averages in ABNM and the trend is similar as we saw in Model 1.0 (Figure 4-3) that people with more contacts have higher chances of getting infected as expected.

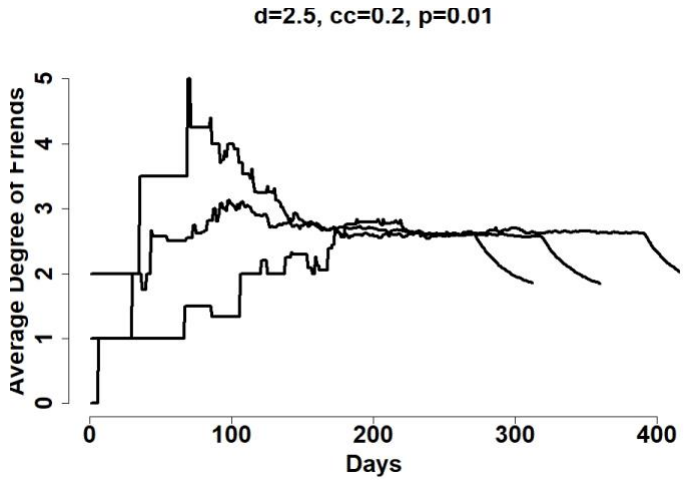


Figure 4-8: Model 1.1 the convergence of average degrees of friend's contact

4.2.2 Computation Time

We compared the computation time of a single simulation until 90 percent of the population is infected on different networks sizes from 300 to 150,000 with transmission probability $p = 0.01$ using both ECNA Model 1.1 and ABNM (NetLogo). Figure 4-9 shows that computation time of ECNA in Java (Model 1.1) is considerably faster than ECNA in MATLAB (Model 1.0) and significantly faster than ABNM in NetLogo. To ensure computational efficiency of Model 1.1, we ran both the ECNA model and ABNM on larger networks and compared the computation time. Figure 4-10 shows that the computation time in ABNM in NetLogo increases much faster as population size increases compare to the ECNA model.

The largest population size example that we used here is 150,000, and the ABNM computation time is 573 minutes (~9 hours) whereas the ECNA model performs the simulation in 32 mins, which is almost 20 times faster.

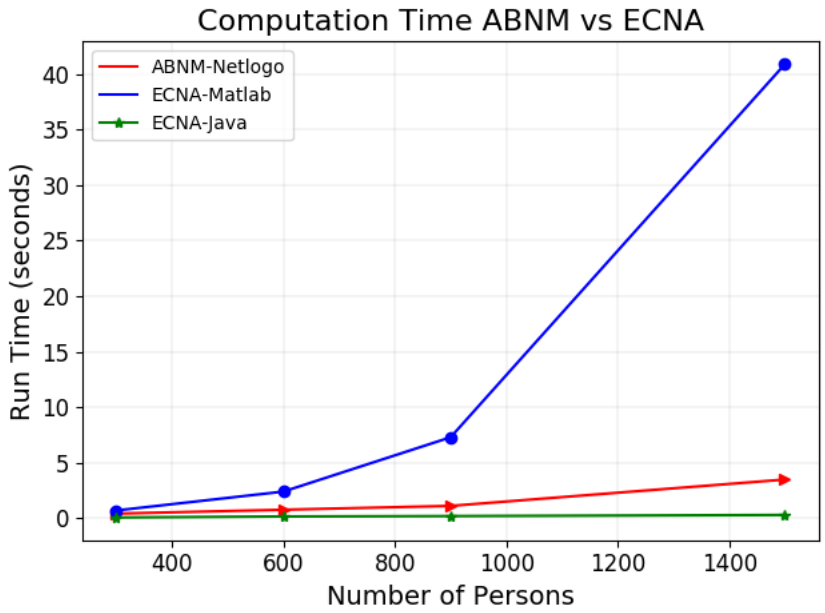


Figure 4-9: Computation time comparison of ECNA (Model 1.0), ECNA (Model 1.1) and ABNM (NetLogo)

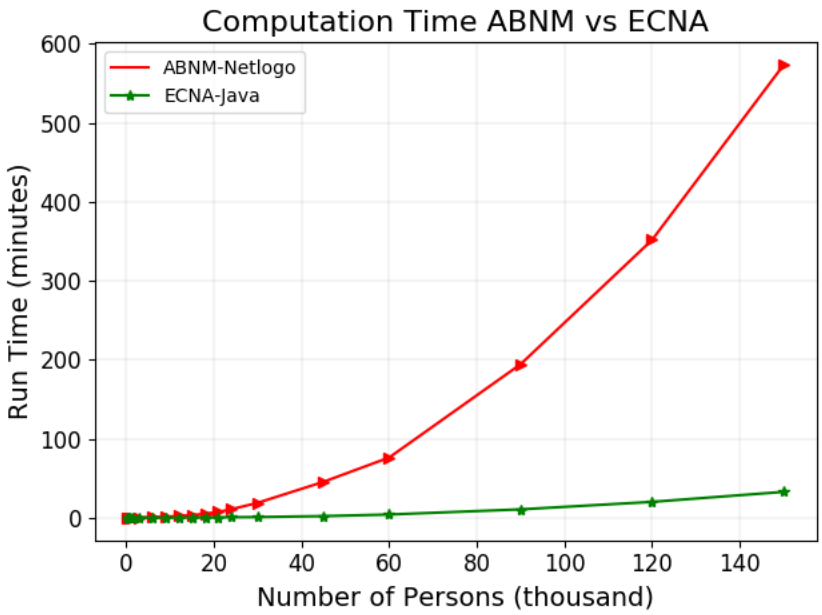


Figure 4-10: Computation time comparison of ECNA (Model 1.1) and ABNM (NetLogo)

4.3 Model 2.0

To test the flexibility of the ECNA in Model 2.0, we generated a network with two communities, where each followed power-law, in a population of 400 persons and simulated the spread of hypothetical diseases using the ECNA and ABNM. Although the core of the ECNA does not change, i.e., it generates only infected contacts, and their contacts as the infection spread over the network, in Model 2.0, there were two significant difference in Model 2.0 than the previous two models.

1. The network consists of two community groups, where each follows power-law, with mixing. Therefore, new parameters for mixing were added to the algorithm.

2. It integrates the neural network model, which was developed in the previous study (Eden et al. 2018), for predicting the distribution of degrees of neighbors of newly infected person so that it can generate a network that follows power-law.

Considering those difference in Model 2.0 from the previous two models, the algorithm and data collection method was modified accordingly, which led to longer computation time when calculating more data. Therefore we simulated the epidemic on the small size of networks to test Model 2.0 in this thesis. However, this computational inefficiency problem can be improved with more time.

4.3.1 Epidemic projection

Figure 4-11 shows a result from 10 simulations of a hypothetical population of 400 persons that consists of two community groups, in which 10 percent mixing between groups, using the ECNA and ABNM, their 5th, 50th and 95th percentiles of the total number of infections at each time step t. The stopping criteria of this simulation were 0.2 that the simulation stops when 20 percent of the population becomes infected. It shows those epidemic projections by the ECNA result similar to ABNM.

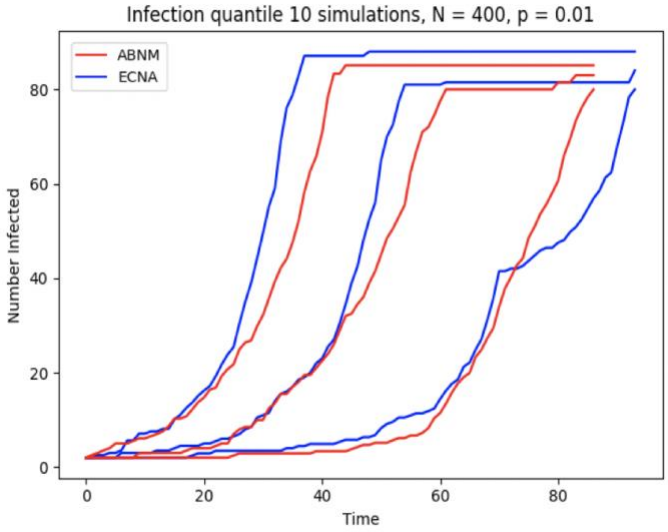


Figure 4-11: Epidemic projection comparison of the ECNA (Model 2.0) and ABNM. 5th, 50th, and 95th percentiles of 10 simulations.

4.3.2 Graph Visualization

Figure 4-12 shows how the network is being evolved using the ECNA, and it was plotted using draw functions in NetworkX. Light blue and pink colors represent uninfected persons and undiscovered person if no link connects to it, and deep blue and red colors represent infected persons.

These graphs show that Model 2.0 successfully generates a network with two community types, where each follows power-law, while simultaneously simulating the transmission of infection over the population using the ECNA and integrating the neural network model.

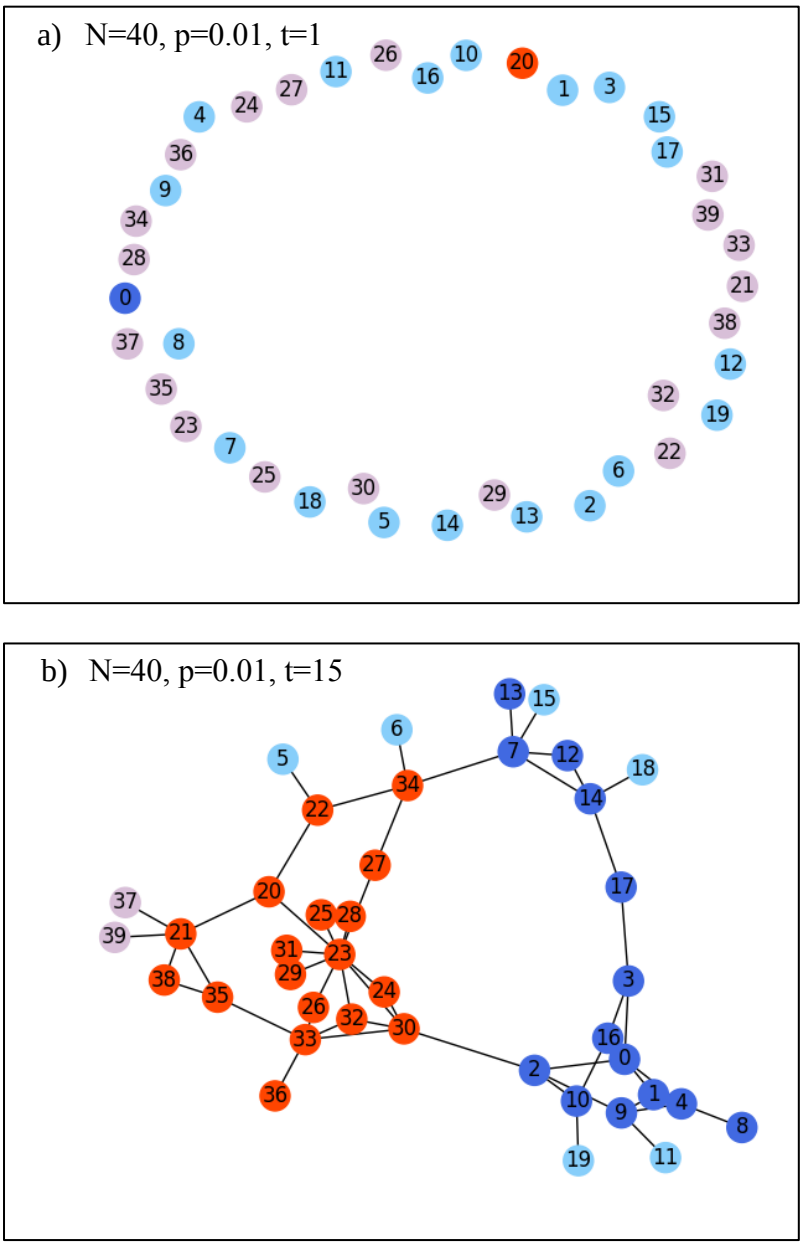


Figure 4-12: An evolution of network with two-communities using the Evolving Contact Network Algorithm. a) An infected contact network at the initial stage of the outbreak b) An infected contact network where the epidemic spread over. Light blue and pink colors represent uninfected persons, deep blue and red colors represent infected persons in the network.

CHAPTER 5

5 CONCLUSIONS

5.1 Summary

This thesis presents different implementations of a new simulation modeling technique, the ECNA, which combines the compartmental model and agent-based modeling techniques, for predicting infectious disease spread. The ECNA is primarily designed for simulating the diseases that transmit through close-contact and have a low prevalence. The network generating mechanism of the ECNA is generating only infected persons and their contacts while simultaneously simulating the spread of the disease.

The objective of this thesis is developing a computationally efficient implementation of ECNA while validating its accuracy of predicting the epidemic and the flexibility of capturing the characteristics of the network as well as an outbreak in different contact network structure settings.

The ECNA is expected to be computational efficient over traditional agent-based models where it requires a full network before simulating the disease transmissions. The computational efficiency of the algorithm provides multiple advantages when simulating epidemic projections whether it is for intervention response during the initial stage of an outbreak or studying the spread of a disease that has a low-prevalence.

This thesis serves as a preliminary proof of concept testing of the new ECNA algorithm, highlighting the promise and significance for more research in this type of modeling.

5.2 Discussion

This thesis features three different models, Model 1.0, Model 1.1, and Model 2.0, that utilizes the ECNA for simulating disease progression.

Model 1.0 and Model 1.1 uses a configuration model mechanism, where degrees of each node is known before the simulation starts, to generate the infected contact network. However, Model 1.1 is an upgrade of Model 1.0 with an improvement of computational efficiency which obtained by using the Adjacency List representation of a graph instead of the Adjacency Matrix representation of a graph. The network structures of the Model 1.0 and Model 1.1 consist of two types of contacts with high clustering coefficients.

Model 2.0 uses a preferential attachment mechanism, which results in a scale-free network, to generate the infected contact network. This model shows an application of the ECNA in generating a network with communities.

Model 1.0 was developed in MATLAB and used an Adjacency Matrix representation of graph to test the accuracy of the ECNA, whereas Model 1.1 was developed in Java and used an Adjacency List representation of graph to test the computational efficiency of the ECNA. These two models were compared to the same ABNM that was developed and simulated in NetLogo. When the computation time of the ECNA in Model 1.0 and ABNM was compared, even in a small size of networks, the computation time of the Model 1.0 was significantly higher than ABNM due to the Adjacency Matrix representation of the networks and the operations on them. However, Model 1.0 provided proper numerical validation on the accuracy of the model so that we were confident to implement the algorithm using a different data structure to improve the computational efficiency in the next model.

Model 1.1 was implemented using *HashSet* class in Java, which has constant time performance when getting and setting data, to represent a graph with Adjacency List. This implementation improved the computation time of the ECNA. The numerical results of the computation time of simulations on the networks of different sizes from 300 to 150,000 nodes showed that ECNA performs significantly faster. However, we compared the computation time of both the ECNA and ABNM until 90 percent of the population was infected. In practice, as ECNA simulates only infected people and their immediate contacts, the population size for ECNA is dramatically smaller than the full population simulated by ABNM, amplifying a greater computational advantage of ECNA over ABNM for simulating low prevalence infectious diseases.

The last model, Model 2.0, presents an implementation of the ECNA for a network that consists of two-community groups, where each community is a scale-free network, using the neural network model that was trained to predict the cumulative distribution of degrees of neighbors of an infected person. Model 2.0 was developed in Python and used the NetworkX package, which provides a *hashable* Graph object, to test the flexibility of the ECNA by applying to different a network with a different structure. This model was compared to an ABNM that was developed and simulated in Python. Python was primarily chosen for Model 2.0 because of the NetworkX package which offers different data structures for representing many types of graphs or network, graph operators and graph generators whereas every function is required to be implemented when using MATLAB and Java in the previous models.

The motivation for Model 2.0 was simulating infectious diseases where community structure or behavior is essential such as different sexual behaviors, homosexual and

heterosexual, in HIV transmission. Based on literature reviews, the real networks including sexual contact networks are scale-free networks, and the preferential attachment is used to generate scale-free networks. However, the preferential attachment cannot be used to ECNA because the degrees of the infected node and their neighbors should be known as soon as the node becomes infected. Therefore, the neural network model was trained on data from epidemic projections on multiple scale-free networks in the previous study and this model was integrated into this Model 2.0 as an empirical validation. This expansion of the ECNA in Model 2.0 successfully projected the epidemic in a small network.

The contribution of the computationally efficient implementation of the algorithm ensures that the ECNA is suitable for simulating the disease that has a low-prevalence in a large population because it eliminates the process of generating the full contact network before simulating the infection. Knowing the ECNA performs faster than traditional ABNM for simulating the disease with low-prevalence allows researchers to study the dynamic of such diseases, where numerous iterations of simulations are required, in a significantly shorter simulation time.

5.3 Limitations

This study has several limitations. For simplicity purposes, we tested the validity of ECNA models using simple disease progression on simple hypothetical networks. The assumptions in simple disease progression include no death and no recovery, a fixed population size during the simulation period and hypothetical disease transmission rate of 0.1 and 0.5 in Model 1.0 and 0.01 in Model 1.1 and Model 2.0. However, the latter transmission rate was based on HIV transmission risk data (Patel et al. 2014). The assumptions in simple hypothetical networks include a fixed hypothetical household size

of three in Model 1.0 and Model 1.1, hypothetical networks with a mixture of 10% between communities in Model 2.0. However, the results of this study suggest that the epidemic projection using the ECNA is similar to ABNM projections using these hypothetical networks and recommend broader expansion of these models using real networks and disease data.

5.4 Conclusion

The need to perform a simulation with shorter computation time required a new algorithm when simulating the spread of diseases that transmit through direct contact and have a low-prevalence.

The purpose of this thesis was to implement a novel algorithm, the Evolving Contact Network Algorithm (ECNA), for predicting the spread of infectious diseases that transmit through close-contact and do empirical analysis on accuracy, computational efficiency, and flexibility of the algorithm.

This thesis presents three different implementations of the ECNA using three different programming languages to test the original hypotheses. Model 1.0 implemented the ECNA in MATLAB and confirmed the accuracy of the algorithm using small size hypothetical networks where an individual can have two types of contacts. Model 1.1 implemented the same algorithm in Java, and this implementation highlighted the computational efficiency of efficient data structures of the Graph object. The underlying algorithm used in Model 1.0 and Model 1.1 was the same and aimed to generate a random graph with two types of contacts, family and friends, each with predefined degree distributions as in configuration models. Finally, Model 2.0 implemented the ECNA using

NetworkX package in Python and tested the flexibility of the algorithm for expansion to two-communities with degree distribution in each network following a power-law.

The key contributions of this thesis are a computationally efficient implementation of the ECNA using Object-Oriented Programming as well as the empirical validation of its accuracy and flexibility of being applied to networks with different structural properties. The contribution of the computationally efficient implementation that uses Adjacency List data structure in Java will have a significant impact in the future studies of the spread of diseases, where contact structures are important and have a low prevalence. This implementation of the algorithm ensures that the ECNA is suitable for simulating epidemic projections whether it is for intervention response during at the initial stage of an outbreak or studying the spread of a disease that has a low-prevalence. Having the implementation of the algorithm developed in Java will be convenient to integrate it with other Java-based softwares such as MASON and Repast that provides agent-based simulation environments.

Moreover, the ECNA models, in this thesis, for generating both random and non-random graphs using prescribed degrees (compartmental model) and determining degrees of contacts while generating the infected persons and their contacts (neural network model) provides promising results for future research in this area for studying further extensions. Future work should consider testing of this method for epidemic projections on real networks.

6 SUPPLEMENT MATERIALS

The source codes are available upon request in the following GitHub repositories. Sample codes for Graph object representations that are used in the models are included in Appendix.

Model 1.0: https://github.com/Buyannemekh/Matlab-MECN
Model 1.1: https://github.com/Buyannemekh/mecn
Model 2.0: https://github.com/Buyannemekh/MECN_py

APPENDIX

THE SOFTWARE CODE FOR DIFFERENT IMPLEMENTATIONS OF GRAPH OBJECT

Model 1.0 code snippet shows Adjacency Matrix representation of graph. The class Contact, which contains information of contact matrix and more, represents a graph in the MATLAB code. The *populate* function, which connects to contacts, was presented as example in the code.

A.1: Model 1.0 in MATLAB

```
classdef Contact<handle
    % A class represents a specific type of Contact
    % family, friends, etc
    properties
        ContactMatrix % A matrix of contacts at the current time
        M % Total number people
    end

    %% methods
    methods
        %% initialization
        function obj = Contact(M)
         obj.M = M;
         obj.ContactMatrix = zeros(M,M);
        end
         %% Make people i,j contacted
        function obj = populate(obj, i, j)
           % i has to be not equal to j
           % assert(~isequal(i,j));
            if (~isequal(i,j))
              obj.ContactMatrix(i,j) = 1;
              obj.ContactMatrix(j,i) = 1;
            end
        end
    end
```

Model 1.1 code snippet shows Adjacency List representation of graph. The class AdjacencyListContact, which extends abstract class called Contact, represents the graph object in the Java code. The *connect* function, which connects to contacts, is presented as example in the code.

A.2: Model 1.1 in Java

```java
public class AdjacencyListContact extends Contact {
    /**
     * List of LinkedList represents adjacency list
     */
    private List<Set<Integer>> adjacencyList;

    /**
     * List of HashSet
     * Each element of the List represents an individual person
     * Each HashSet of i th element of the list is a set of contacts of i
     * @param numberOfPeople
     * @param type
     */
    public AdjacencyListContact (int numberOfPeople, ContactType type) {
        super(numberOfPeople, type);
        this.adjacencyList = new ArrayList<Set<Integer>>(numberOfPeople);

        //create adjacency list for every person
        for (int ii=0; ii<numberOfPeople; ii++) {
            adjacencyList.add(new HashSet<Integer>());
        }
    }

    @Override
    public void connect(int i, int j) {
        if (i == j) {
            throw new IllegalArgumentException(
                    "Person cannot be connected to itself");
        }
        adjacencyList.get(i).add(j);
        adjacencyList.get(j).add(i);
    }
```

Model 2.0 code snippet shows Graph object in Python NetworkX package. The in-built *add_edge* function, which connects to contacts, is presented as example in the code.

A.3: Model 2.0 in Python

```
import networkx as nx

G = nx.empty_graph(n=num_nodes) # initialize a graph

if not index_of_discovered == index_of_infected:
    G.add_edge(index_of_infected, index_of_discovered)
```

BIBLIOGRAPHY

Anderson, Roy M., and Robert M. May. 1991. *Infectious Diseases of Humans: Dynamics and Control*. OUP Oxford.

Andre, McKenzie, Kashef Ijaz, Jon D. Tillinghast, Valdis E. Krebs, Lois A. Diem, Beverly Metchock, Theresa Crisp, and Peter D. McElroy. 2011. "Transmission Network Analysis to Complement Routine Tuberculosis Contact Investigations." Research-article. American Journal of Public Health. October 10, 2011. https://doi.org/10.2105/AJPH.2005.071936.

Bajardi, Paolo, Chiara Poletto, Jose J. Ramasco, Michele Tizzoni, Vittoria Colizza, and Alessandro Vespignani. 2011. "Human Mobility Networks, Travel Restrictions, and the Global Spread of 2009 H1N1 Pandemic." *PLOS ONE* 6 (1): e16591. https://doi.org/10.1371/journal.pone.0016591.

Balcan, Duygu, Bruno Gonçalves, Hao Hu, José J. Ramasco, Vittoria Colizza, and Alessandro Vespignani. 2010. "Modeling the Spatial Spread of Infectious Diseases: The GLobal Epidemic and Mobility Computational Model." *Journal of Computational Science* 1 (3): 132–45. https://doi.org/10.1016/j.jocs.2010.07.002.

Barabasi, Albert-Laszlo, and Reka Albert. 1999. "Emergence of Scaling in Random Networks" 286: 5.

Barrett, Christopher L., Keith R. Bisset, Stephen G. Eubank, Xizhou Feng, and Madhav V. Marathe. 2008. "EpiSimdemics: An Efficient Algorithm for Simulating the Spread of Infectious Disease over Large Realistic Social Networks." In *2008 SC - International Conference for High Performance Computing, Networking, Storage and Analysis*, 1–12. Austin, TX, USA: IEEE. https://doi.org/10.1109/SC.2008.5214892.

Bender, Edward A, and E.Rodney Canfield. 1978. "The Asymptotic Number of Labeled Graphs with given Degree Sequences." *Journal of Combinatorial Theory, Series A* 24 (3): 296–307. https://doi.org/10.1016/0097-3165(78)90059-6.

Chung, F., and L. Lu. 2002. "The Average Distances in Random Graphs with given Expected Degrees." *Proceedings of the National Academy of Sciences* 99 (25): 15879–82. https://doi.org/10.1073/pnas.252631999.

Centers for Disease Control and Prevention. Estimated HIV incidence in the United States, 2007– 2010. HIV Surveillance Supplemental Report 2012;17(No. 4). http://www.cdc.gov/hiv/topics/ surveillance/resources/reports/#supplemental. Published December 2012. Accessed [March, 2019].

"Creating, Concatenating, and Expanding Matrices - MATLAB & Simulink." n.d. Accessed March 20, 2019. https://www.mathworks.com/help/matlab/math/creating-and-concatenating-matrices.html.

Diekmann, O., and J. A. P. Heesterbeek. 2000. *Mathematical Epidemiology of Infectious Diseases: Model Building, Analysis and Interpretation*. John Wiley & Sons.

Eubank, Stephen, Hasan Guclu, V. S. Anil Kumar, Madhav V. Marathe, Aravind Srinivasan, Zoltán Toroczkai, and Nan Wang. 2004. "Modelling Disease Outbreaks in Realistic Urban Social Networks." *Nature* 429 (6988): 180–84. https://doi.org/10.1038/nature02541.

Eden, Matthew, Rebecca Castonguay, Buyannemekh Munkhbat, Hari Balasubramanian, and Chaitra Gopalappa. 2018. "Evolving contact network algorithm: a new

simulation method for modeling HIV, a disease with low prevalence but a critical public health issue in the US". *Under Review*.

Ferguson, Neil M., Matt J. Keeling, W. John Edmunds, Raymond Gani, Bryan T. Grenfell, Roy M. Anderson, and Steve Leach. 2003. "Planning for Smallpox Outbreaks." *Nature* 425 (6959): 681–85. https://doi.org/10.1038/nature02007.

Goodreau, Steven M., Nicole B. Carnegie, Eric Vittinghoff, Javier R. Lama, Jorge Sanchez, Beatriz Grinsztejn, Beryl A. Koblin, Kenneth H. Mayer, and Susan P. Buchbinder. 2012. "What Drives the US and Peruvian HIV Epidemics in Men Who Have Sex with Men (MSM)?" *PLoS ONE*. https://doi.org/10.1371/journal.pone.0050522.

Hamill, Lynne, and Nigel Gilbert. 2009. "Social Circles: A Simple Structure for Agent-Based Social Network Models," 23.

Keeling, Matt J., and Ken T.D. Eames. 2005. "Networks and Epidemic Models." *Journal of The Royal Society Interface* 2 (4): 295–307. https://doi.org/10.1098/rsif.2005.0051.

Kermack, William Ogilvy, and A. G. McKendrick. 1927. "Contributions to the Mathematical Theory of Epidemics. II. —The Problem of Endemicity." *Proc. R. Soc. Lond. A* 138 (834): 55–83. https://doi.org/10.1098/rspa.1932.0171.

Le Polain de Waroux, Oliver, J. Kucharski, A, Juan-Giner, A, Flasche, S, Tumwesigye, E, Arinaitwe, R, Mwanga, Juliet, et al. 2018. "Characteristics of Human Encounters and Social Mixing Patterns Relevant to Infectious Diseases Spread by Close Contact: A Survey in Southwest Uganda." *BMC Infectious Diseases* 18 (10.1186/s12879-018-3073-1.).

Longini, I. M. 2005. "Containing Pandemic Influenza at the Source." *Science* 309 (5737): 1083–87. https://doi.org/10.1126/science.1115717.

"Matrices and Arrays - MATLAB & Simulink." n.d. Accessed March 19, 2019. https://www.mathworks.com/help/matlab/learn_matlab/matrices-and-arrays.html

Adams, Ben. 2016. "Household Demographic Determinants of Ebola Epidemic Risk." *Journal of Theoretical Biology* 392 (March): 99–106. https://doi.org/10.1016/j.jtbi.2015.11.025.

Ajelli, Marco, Bruno Gonçalves, Duygu Balcan, Vittoria Colizza, Hao Hu, José J. Ramasco, Stefano Merler, and Alessandro Vespignani. 2010. "Comparing Large-Scale Computational Approaches to Epidemic Modeling: Agent-Based versus Structured Metapopulation Models." *BMC Infectious Diseases* 10 (1): 190. https://doi.org/10.1186/1471-2334-10-190.

Alexander, Kathleen A., Claire E. Sanderson, Madav Marathe, Bryan L. Lewis, Caitlin M. Rivers, Jeffrey Shaman, John M. Drake, et al. 2015. "What Factors Might Have Led to the Emergence of Ebola in West Africa?" *PLOS Neglected Tropical Diseases* 9 (6): e0003652. https://doi.org/10.1371/journal.pntd.0003652.

Bansal Shweta, Grenfell Bryan T, and Meyers Lauren Ancel. 2007. "When Individual Behaviour Matters: Homogeneous and Network Models in Epidemiology." *Journal of The Royal Society Interface* 4 (16): 879–91. https://doi.org/10.1098/rsif.2007.1100.

Barabási, Albert-László, Erzsébet Ravasz, and Tamás Vicsek. 2001. "Deterministic Scale-Free Networks." *Physica A: Statistical Mechanics and Its Applications* 299 (3): 559–64. https://doi.org/10.1016/S0378-4371(01)00369-7.

Bauch, Chris T., and Alison P. Galvani. 2013. “Social Factors in Epidemiology.” *Science* 342 (6154): 47–49. https://doi.org/10.1126/science.1244492.

Berryman, Matthew. 2008. “Review of Software Platforms for Agent Based Models.” DSTO-GD-0532. DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION EDINBURGH (AUSTRALIA) LAND OPERATIONS DIV. https://apps.dtic.mil/docs/citations/ADA485784.

Berryman, Matthew J., and Simon D. Angus. 2010. “Tutorials on Agent-Based Modelling with NetLogo and Network Analysis with Pajek.” In *Complex Physical, Biophysical and Econophysical Systems*, 351–75. The Australian National University, Canberra: WORLD SCIENTIFIC. https://doi.org/10.1142/9789814277327_0010.

Cauchemez, Simon, Christl A. Donnelly, Carrie Reed, Azra C. Ghani, Christophe Fraser, Charlotte K. Kent, Lyn Finelli, and Neil M. Ferguson. 2009. “Household Transmission of 2009 Pandemic Influenza A (H1N1) Virus in the United States.” *New England Journal of Medicine* 361 (27): 2619–27. https://doi.org/10.1056/NEJMoa0905498.

Center for Disease Control. 2016. “2014 Ebola Outbreak in West Africa Epidemic Curves | 2014-2016 Outbreak West Africa.” https://www.cdc.gov/vhf/ebola/history/2014-2016-outbreak/cumulative-cases-graphs.html.

Coburn, Brian J., Bradley G. Wagner, and Sally Blower. 2009. “Modeling Influenza Epidemics and Pandemics: Insights into the Future of Swine Flu (H1N1).” *BMC Medicine* 7 (1): 30. https://doi.org/10.1186/1741-7015-7-30.

Cohen, Mitchell L. 2000. “Changing Patterns of Infectious Disease.” *Nature* 406 (August): 762–67. https://doi.org/10.1038/35021206.

“Creating, Concatenating, and Expanding Matrices - MATLAB & Simulink.” n.d. Accessed March 20, 2019. https://www.mathworks.com/help/matlab/math/creating-and-concatenating-matrices.html.

Danon, Leon, Ashley P. Ford, Thomas House, Chris P. Jewell, Matt J. Keeling, Gareth O. Roberts, Joshua V. Ross, and Matthew C. Vernon. 2011. “Networks and the Epidemiology of Infectious Disease.” Research article. Interdisciplinary Perspectives on Infectious Diseases. 2011. https://doi.org/10.1155/2011/284909.

“Data for Sierra Leone, Liberia | Data.” n.d. Accessed March 20, 2019. https://data.worldbank.org/?locations=SL-LR.

Derouich, M., A. Boutayeb, and EH Twizell. 2003. “A Model of Dengue Fever.” *BioMedical Engineering OnLine* 2 (1): 4. https://doi.org/10.1186/1475-925X-2-4.

Drake, John M., Iurii Bakach, Matthew R. Just, Suzanne M. O’Regan, Manoj Gambhir, and Isaac Chun-Hai Fung. 2015. “Transmission Models of Historical Ebola Outbreaks.” *Emerging Infectious Diseases* 21 (8): 1447–50. https://doi.org/10.3201/eid2108.141613.

Enright, Jessica, and Rowland Raymond Kao. 2018. “Epidemics on Dynamic Networks.” *Epidemics* 24 (September): 88–97. https://doi.org/10.1016/j.epidem.2018.04.003.

Eubank, Stephen, Hasan Guclu, V. S. Anil Kumar, Madhav V. Marathe, Aravind Srinivasan, Zoltán Toroczkai, and Nan Wang. 2004. “Modelling Disease

Outbreaks in Realistic Urban Social Networks." *Nature* 429 (6988): 180–84. https://doi.org/10.1038/nature02541.

Fotouhi, Babak, and Michael G. Rabbat. 2013. "Degree Correlation in Scale-Free Graphs." *The European Physical Journal B* 86 (12). https://doi.org/10.1140/epjb/e2013-40920-6.

Hagberg, Aric, Pieter Swart, and Daniel S Chult. 2008. "Exploring Network Structure, Dynamics, and Function Using Networkx." LA-UR-08-05495; LA-UR-08-5495. Los Alamos National Lab. (LANL), Los Alamos, NM (United States). https://www.osti.gov/biblio/960616.

Hethcote, Herbert W. 1994. "A Thousand and One Epidemic Models." In *Frontiers in Mathematical Biology*, edited by Simon A. Levin, 504–15. Lecture Notes in Biomathematics. Springer Berlin Heidelberg.

Jiang, Lurong, Xinyu Jin, Yongxiang Xia, Bo Ouyang, and Duanpo Wu. 2014. "Dynamic Behavior of the Interaction between Epidemics and Cascades on Heterogeneous Networks." *EPL (Europhysics Letters)* 108 (5): 58009. https://doi.org/10.1209/0295-5075/108/58009.

Keeling, Matt. 2005. "The Implications of Network Structure for Epidemic Dynamics." *Theoretical Population Biology* 67 (1): 1–8. https://doi.org/10.1016/j.tpb.2004.08.002.

Liljeros, Fredrik, Christofer R. Edling, Luís A. Nunes Amaral, H. Eugene Stanley, and Yvonne Åberg. 2001. "The Web of Human Sexual Contacts." *Nature* 411 (6840): 907–8. https://doi.org/10.1038/35082140.

Merli, M. Giovanna, and Sara Hertog. 2010. "Masculine Sex Ratios, Population Age Structure and the Potential Spread of HIV in China." *Demographic Research* 22: 63–94.

Newman, Mark. 2010. *Networks: An Introduction*. Oxford University Press.

Potterat, J. J., L. Phillips-Plummer, S. Q. Muth, R. B. Rothenberg, D. E. Woodhouse, T. S. Maldonado-Long, H. P. Zimmerman, and J. B. Muth. 2002. "Risk Network Structure in the Early Epidemic Phase of HIV Transmission in Colorado Springs." *Sexually Transmitted Infections* 78 (suppl 1): i159–63. https://doi.org/10.1136/sti.78.suppl_1.i159.

Rocha, Luis E. C., Fredrik Liljeros, and Petter Holme. 2011. "Simulated Epidemics in an Empirical Spatiotemporal Network of 50,185 Sexual Contacts." Edited by Marcel Salathé. *PLoS Computational Biology* 7 (3): e1001109. https://doi.org/10.1371/journal.pcbi.1001109.

Sargent, R. G. 2010. "Verification and Validation of Simulation Models." In *Proceedings of the 2010 Winter Simulation Conference*, 166–83. https://doi.org/10.1109/WSC.2010.5679166.

Schneeberger, Anne, Catherine H. Mercer, Simon a. J. Gregson, Neil M. Ferguson, Constance A. Nyamukapa, Roy M. Anderson, Anne M. Johnson, and Geoff P. Garnett. 2004. "Scale-Free Networks and Sexually Transmitted Diseases: A Description of Observed Patterns of Sexual Contacts in Britain and Zimbabwe." *Sexually Transmitted Diseases* 31 (6): 380.

Siettos, Constantinos, Cleo Anastassopoulou, Lucia Russo, Christos Grigoras, and Eleftherios Mylonakis. 2015. "Modeling the 2014 Ebola Virus Epidemic – Agent-Based Simulations, Temporal Analysis and Future Predictions for Liberia and

Sierra Leone." *PLoS Currents* 7 (March). https://doi.org/10.1371/currents.outbreaks.8d5984114855fc425e699e1a18cdc6c9.
Suryaprasad, Anil, John T. Redd, Kathy Hancock, Alicia Branch, Evelene Steward-Clark, Jacqueline M. Katz, Alicia M. Fry, and James E. Cheek. 2013. "Severe Acute Respiratory Infections Caused by 2009 Pandemic Influenza A (H1N1) among American Indians—Southwestern United States, May 1–July 21, 2009." *Influenza and Other Respiratory Viruses* 7 (6): 1361–69. https://doi.org/10.1111/irv.12123.
Volz, Erik M., Joel C. Miller, Alison Galvani, and Lauren Ancel Meyers. 2011. "Effects of Heterogeneous and Clustered Contact Patterns on Infectious Disease Dynamics." *PLOS Computational Biology* 7 (6): e1002042. https://doi.org/10.1371/journal.pcbi.1002042.
"Welcome to ComplexNetworkSim's Documentation! — ComplexNetworkSim v0.1.2 Documentation." n.d. Accessed March 21, 2019. https://pythonhosted.org/ComplexNetworkSim/.
Widgren, Stefan, Pavol Bauer, Robin Eriksson, and Stefan Engblom. 2016. "SimInf: An R Package for Data-Driven Stochastic Disease Spread Simulations." *ArXiv:1605.01421 [q-Bio, Stat]*, May. http://arxiv.org/abs/1605.01421.
Willem, Lander, Frederik Verelst, Joke Bilcke, Niel Hens, and Philippe Beutels. 2017. "Lessons from a Decade of Individual-Based Models for Infectious Disease Transmission: A Systematic Review (2006-2015)." *BMC Infectious Diseases* 17 (1). https://doi.org/10.1186/s12879-017-2699-8.

Molloy, Michael, and Bruce Reed. 1995. "A Critical Point for Random Graphs with a given Degree Sequence." *Random Structures & Algorithms* 6 (2–3): 161–80. https://doi.org/10.1002/rsa.3240060204.
Mossong, Joël, Niel Hens, Mark Jit, Philippe Beutels, Kari Auranen, Rafael Mikolajczyk, Marco Massari, et al. 2008. "Social Contacts and Mixing Patterns Relevant to the Spread of Infectious Diseases." Edited by Steven Riley. *PLoS Medicine* 5 (3): e74. https://doi.org/10.1371/journal.pmed.0050074.
Newman, M. E. J. 2006. "The Structure and Function of Complex Networks." *SIAM Review*, August. https://doi.org/10.1137/S003614450342480.
Pastor-Satorras, Romualdo, Claudio Castellano, Piet Van Mieghem, and Alessandro Vespignani. 2015. "Epidemic Processes in Complex Networks." *Reviews of Modern Physics* 87 (3): 925–79. https://doi.org/10.1103/RevModPhys.87.925.
Paul Erdős, Alfréd Rényi. 1960. "On the Evolution of Random Graphs." *Publicationes Mathematicae* 290 (6).
Paulinevan den Driessche. 2017. "Reproduction Numbers of Infectious Disease Models." *Infectious Disease Modelling* 2 (3): 288–303. https://doi.org/10.1016/j.idm.2017.06.002.
Poletto, C., M. F. Gomes, A. Pastore y Piontti, L. Rossi, L. Bioglio, D. L. Chao, I. M. Longini Jr, M. E. Halloran, V. Colizza, and A. Vespignani. 2014. "Assessing the Impact of Travel Restrictions on International Spread of the 2014 West African Ebola Epidemic." *Eurosurveillance* 19 (42): 20936. https://doi.org/10.2807/1560-7917.ES2014.19.42.20936.
Rahmandad, Hazhir, and John Sterman. 2008. "Heterogeneity and Network Structure in the Dynamics of Diffusion: Comparing Agent-Based and Differential Equation

Models." *Management Science* 54 (5): 998–1014. https://doi.org/10.1287/mnsc.1070.0787.
Read, Jonathan M., Ken T. D. Eames, and W. John Edmunds. 2008. "Dynamic Social Networks and the Implications for the Spread of Infectious Disease." *Journal of The Royal Society Interface* 5 (26): 1001–7. https://doi.org/10.1098/rsif.2008.0013.
Sergey Dorogovtsev. 2010. *Lectures on Complex Networks*. Oxford University Press, Oxford.
Simon, Péter L., Michael Taylor, and Istvan Z. Kiss. 2011. "Exact Epidemic Models on Graphs Using Graph-Automorphism Driven Lumping." *Journal of Mathematical Biology* 62 (4): 479–508. https://doi.org/10.1007/s00285-010-0344-x.
Smieszek, Timo, Lena Fiebig, and Roland W. Scholz. 2009. "Models of Epidemics: When Contact Repetition and Clustering Should Be Included." *Theoretical Biology and Medical Modelling* 6 (1). https://doi.org/10.1186/1742-4682-6-11.
Tizzoni, Michele, Paolo Bajardi, Chiara Poletto, José J. Ramasco, Duygu Balcan, Bruno Gonçalves, Nicola Perra, Vittoria Colizza, and Alessandro Vespignani. 2012. "Real-Time Numerical Forecast of Global Epidemic Spreading: Case Study of 2009 A/H1N1pdm." *BMC Medicine* 10 (1): 165. https://doi.org/10.1186/1741-7015-10-165.
Willem, Lander, Frederik Verelst, Joke Bilcke, Niel Hens, and Philippe Beutels. 2017. "Lessons from a Decade of Individual-Based Models for Infectious Disease Transmission: A Systematic Review (2006-2015)." *BMC Infectious Diseases* 17 (1). https://doi.org/10.1186/s12879-017-2699-8.
Wilson, Robin J. 1998. *Introduction to Graph Theory*. 4. ed., [Nachdr.]. Harlow: Prentice Hall.