”You are never further than one decision away from making a difference. It doesn’t matter whether it’s a big difference, doesn’t matter if it was a small difference, because you don’t have to save the world by yourself. In fact you can’t. All you have to do is lay down one brick. All you have to do is make things a little bit better in a small way so other people can lay their brick on top of that, or beside that. And together, step-by-step, day-by-day, year-by-year, we build the foundation of something better...We make things better, we become safe, together, right. Collectively that is our strength. That is the power of civilization. That is the power that shapes the future.”

Edward Snowden

Epidemic Simulation and Mitigation
via Evolutionary Computation

Michael Dubé, BSc

Computer Science

Submitted in partial fulfilment
of the requirements for the degree of

Master of Science

Faculty of Mathematics and Science, Brock University
St. Catharines, Ontario

© 2020

Abstract

A global pandemic remains a public health event that presents a unique and unpredictable challenge for those making health related decisions and the populations who experience the virus. Though a pandemic also provides the opportunity for researchers and health administrations around the world to mobilize in the fields of epidemiology, computer science, and mathematics to generate epidemic models, vaccines, and vaccination strategies to mitigate unfavourable outcomes. To this end, a generative representation to create personal contact networks, representing the social connections within a population, known as the Local THADS-N generative representation is introduced and expanded upon. This representation uses an evolutionary algorithm and is modified to include new local edge operations improving the performance of the system across several test problems. These problems include an epidemic's duration, spread through a population, and closeness to past epidemic behaviour. The system is further developed to represent sub-communities known as districts, better articulating epidemics spreading within and between neighbourhoods. In addition, the representation is used to simulate four competing vaccination strategies in preparation for iterative vaccine deployment amongst a population, an inevitability when considering the lag inherent to developing vaccines. Finally, the Susceptible-Infected-Removed (SIR) model of infection used by the system is expanded in preparation for adding an asymptomatic state of infection as seen within the COVID-19 pandemic.

Acknowledgements

I want to pay special thanks to the people who saw more in me than I saw in myself at the time. They helped me to aim higher, think bigger, and achieve more than I thought I could motivating, inspiring, and believing in me. This list is extensive so I will highlight those who featured most prominently.

First and foremost is my supervisor, Sheridan, who first saw within me research potential and asked me to give it a try. She has always been a voice of encouragement, guidance, and kindness; helping me to remain focused on what needed to be done and getting it done. She is a remarkable steward of the culture within the Computer Science department that is constantly highlighted as an asset at Brock. Thank you for your time and tea breaks, they mean the world to me.

Next, I want to thank Dan who provided the first iteration of the system used herein as well as constant guidance on how to improve the system. I was always interested in epidemic modelling and you provided a vehicle for doing so as well as the odd movie reference.

Thank you also to my thesis committee members Ke and Brian whom helped to support and provide direction throughout my Master's. Your knowledge and experience were an asset.

A shout-out to our system administrator Cale who had to deal with me hogging sandcastle CPU cores to meet paper deadlines. Thank you for your CPU cycles, support, and helping me to complete my degree.

This would be incomplete without thanking my mom, family, and friends that tolerated days or weeks of ghosting when I was approaching deadlines. Your understanding nature helped me complete this work by supporting me and encouraging me along the way. Thank you for the check-ins, support, and spaghetti, without them it would be impastabowl to have made it to the finish line.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Epidemics, such as COVID-19, continue to offer many challenges to human populations as they spread without the possibility of immediate detection and evolve continually. The societies which experience the epidemic must adapt their way of life, after detection, in order to minimize the impact of the virus. This puts an enormous burden on public heath administrations, researchers, and the governmental officials responsible for discovering, tracking, modelling, mitigating, and searching for a vaccine for the virus [47].

This thesis encompasses aspects of epidemic modelling, mitigation, and vaccine deployment using evolutionary algorithms to evolve personal contact networks. Evolutionary algorithms, one branch of evolutionary computation [41], were proposed in [6] to investigate their ability at generating these networks, and have demonstrated their worth. Since then our system has undergone several iterations with each adding new functionality or making refinements. Thus, this work presents a version of the algorithm that aims to consummate this investigation. This includes finalizing the representation used by the algorithm, evaluating a novel parameter selection technique known as *point packing* [5], and exploring the algorithm's ability to generate networks under various epidemic models [2]. In addition, the networks generated by the algorithm are used to test different vaccination strategies and explore the impact sub-communities within a population has on epidemic behaviour. These modifications are elaborated upon and evaluated within the case studies in subsequent chapters.

The personal contact networks generated by the evolutionary algorithm represent the physical connections between members of a population and are the basis of the work completed herein. They are generated using graph induction, and then used to simulate epidemics. These simulated epidemics are evaluated against various test problems in order to improve the representation, develop vaccination strategies, and expand the model of infection to more accurately mirror that seen with COVID-19.

When assessing a particular network there are a number of metrics which can be harnessed to determine its relative utility. The following metrics are deployed in this thesis: how long a virus spreads within the population, known as *epidemic duration*; how close a simulated epidemic is to prior observed epidemic behaviour or *epidemic profile matching*; and *epidemic spread*, which is the number of individuals in a population which become infected with the virus. These metrics are used as a measure of the representation's ability to model the epidemic or a vaccination strategy's effectiveness at mitigating the impact of the epidemic.

1.1 Structure of the Paper

The rest of this thesis is structured in the following manner. Chapter 2 broadly introduces the field of epidemiology including vaccine development and models for simulating epidemics. Chapter 3 outlines evolutionary algorithms, including the decisions that need to be considered when using this tool to solve any number of test problems. Chapter 4 provides a complete overview of the evolutionary algorithm and related components used in the body of work included subsequently. Chapters 5 through 8 are made up of four papers that were published in conferences and utilize said evolutionary algorithm. Finally, Chapter 9 concludes the advances made through iterative improvements to the system and suggests future research potential in this area.

Chapter 2

Background Research

The work completed as part of this Master's thesis relies upon prior work completed by researchers coming from multiple disciplines. The spread of viruses in a human population combines areas of research in mathematics, statistics, biology, medicine, and epidemiology. No single discipline is able to conduct the research necessary to sequence a virus, develop a vaccine or intervention, collect data to generate a personal contact network, monitor and model the spread of the virus, and formulate appropriate and effective social policy to mitigate the scope and degree of harm done by the virus on the population [17].

2.1 Epidemiology

Epidemiology broadly studies the relationship between a population's behaviour or status and the health outcomes for members of the population. The intent of epidemiological research is to apply interventions within populations with the explicit goal of improving a given health outcome. Health authorities around the world utilize epidemiology to analyse the state of their population and make decisions regarding the ailments afflicting the population. When considering the spread of a virus, the total number of infections divided by the number of people living in a particular country or region provides a rate of infection for said region. This is one metric that can be used to compare a virus' intensity between two regions. Though epidemiology relies on a broader set of information from several disciplines in order to form a complete picture of the environment and the relevant characteristics pertaining to the virus [17].

The pattern of infection through a population can be affected by time, locale, and an individual's health. The locale includes factors such as: climate in the region, differences between urban and rural environments, and where infection hubs such as schools are located. An individual's age, sex, socioeconomic status, past medical history, and place of work are some characteristics that can impact the likelihood and severity of infection within an individual. In addition, epidemiologists generate their interventions assuming the incidence of a new disease occurs only when a certain combination of risk factors combines within an individual. These risk factors are referred to as determinants and are the focus of epidemiological research. The risk assessments of each of these determinants informs those making the everyday decisions impacting the spread of a virus. The assessment of risk requires robust health surveillance [17].

The surveillance necessary to track the spread of a virus and the health of a population requires the perpetual, systematic collection, evaluation, integration, and circulation of health data to global, federal, and local health officials. The accuracy, timeliness, and robustness of this data directly impacts the quality and effectiveness of the decisions made by these officials [17]. One example is contact tracing; obtaining a complete record of infection as well as potential infections can prevent an epidemic within a country should the surveillance be of sufficient breadth, depth, and, most critically, speed. Another example is contact networks comprising data on those who come into contact with one-another; affording projections of virus spread through a community. These are valuable assets to those choosing which control measures to implement in response to a global pandemic.

Control measures enacted upon a population can aim to prevent an ailment from arriving within the region, eliminate a virus from the region, or control the spread of a virus through the region. Normally, these measures target those in the chain of transmission which are most receptive to the intervention being considered. The chain of transmission includes: the *agents* within infected members of the population, any other *source(s)* of the virus, how the virus moves from person-to-person or the *mode of transmission*, the *portal of entry* into the body, and the people, or *hosts*, susceptible to becoming infected [17]. The appropriate control measures vary depending on the disease being considered and the environment in which it is spreading, though some common measures exist. Isolating those who are infected could prevent direct transmission of the virus while modifications to social norms such as face coverings, physical distancing, and modified greetings can reduce the likelihood of transmission. Whereas, the development and distribution of a vaccine increases a host's defences by preventing the virus from taking root within the body. Should a critical mass of individuals be immune then herd immunity is established; thus preventing a pathogen from being able to spread through the population by eliminating potential chains of infection. Though, the proportion of immune people within the population necessary to achieve herd immunity varies by disease and is unknown at the outset. These and other control measures are tracked and modified in response to the epidemic curve of a virus. The epidemic curve is generated by plotting the days since the start of an outbreak against the number of new confirmed cases within the population. The epidemic curve provides a useful visualization of the severity of the virus in a population with peaks resembling times of rapid spread and troughs when the virus is being controlled effectively or eradicated from the population.

Every susceptible person who comes into contact with a new virus has the potential to become infected. Those that do become infected undergo an incubation period, known as the *subclinical* period of disease, in which they are unaware they have the virus and may or may not be infectious. The length of the incubation period ranges from a few moments to multiple decades. For some viruses once the ailment has reached a critical mass an individual may become infectious yet asymptomatic, lacking any presence of the contagion in lab tests; these individuals are known as *carriers*. Carriers take few precautions to prevent transmission of the virus as they are unaware they are infected, let alone infectious. Once symptoms appear within an individual they are considered to have entered the *clinical* period of disease. This is when most cases can be confirmed by health professionals. The length of the clinical period exhibits the same variability in length as does the subclinical period. The vast difference between severity of a virus in any particular individual is known as the *spectrum of disease* and is the reason why some remain asymptomatic while other succumb to the virus. Thus, an infected individual will either recover from the virus, be in a state of disability, or pass away [17].

2.2 Vaccination

Vaccines enhance the immune system of a potential host by providing it with genetic material of a disease, allowing for the production of antibodies that target the disease. Antibodies are proteins that neutralize the cells of an invading virus by attaching themselves to the cell membrane of the virus. The cells of the foreign pathogen are known as antigens. The genetic material from the vaccine allows the body to develop an antibody before a potential host comes into contact with the virus [18, 27, 39]. Thus, when the virus is detected within the body the antibody begins being released into the bloodstream. When an antibody for a particular ailment comes into contact with the invading cell, the invader is neutralized. A vaccine essentially allows our body to generate an antibody for a virus before being exposed to said virus. Without a vaccine, an individual will only begin the production of the required antibody after the virus has taken a foothold. This delay allows for the disease to multiply, spread, and manifest as viral symptoms within the body. Should the immune system of an individual fail to formulate the required antibody before the period of infectability of the virus or the onset of symptoms then that individual will be able to pass on the virus to others and experience the symptoms associated with the virus; these symptoms could be mild (i.e. coughing and sneezing), moderate (i.e. requiring admittance to the hospital), or severe (i.e. permanent disability or death). The development of the antibody by the immune system before exposure allows for an immediate response, neutralizing the virus before it can spread to others or cause damage to the body.

The production of a vaccine requires scientists to modify the DNA of a virus such that it won't infect the host and will allow the immune system to generate the required antibody. This requires a substantial amount of trial-and-error, with the discovery of a successful vaccine candidate not guaranteed. In addition, governments and international organizations have formalized the phases necessary to ensure unsafe and ineffective vaccine candidates do not receive approval. Should this occur, public trust in the particular vaccine and vaccination in general would diminish, in turn reducing the available funding to the generation of vaccines in the first place. The phases of vaccine development include: the *exploratory* stage where scientists search for antigens that may trigger the same immune response as the pathogen; the *pre-clinical* stage where a vaccine candidate is tested using tissue samples or with animal studies; *clinical Phase 1* dose-ranging trials on less than 100 individuals who undergo intense observation to determine the safety and dosing schedule of the candidate vaccine; *clinical phase 2* trials with hundreds of test subjects from the vaccine's target population to reduce side effects; *clinical phase 3* trials which seek to ensure the candidate vaccine is both safe and sufficiently effective when administered to thousands of test subjects; the *regulatory review and approval* stage in which the vaccine is measured against regulatory standards and approved by a country's health authority; lastly, the *quality control* stage which ensures that the vaccine continues to remain effective and safe for its life-cycle; this stage may include a fourth clinical phase to ensure the vaccine is sufficiently effective and safe for mass immunization of the general population [27].

Upon the discovery, refinement, and regulatory approval of a vaccine for a disease the global health production capacities must expand to meet the demand for the new vaccine, especially when experiencing a pandemic on a global scale such as COVID-19. The initial demand upon the discovery of a vaccine requires the drug manufacturers to be prepared to begin scaling up production immediately in order to meet the initial demand of the new vaccine; especially during a pandemic. The manufacture of vaccine can be impeded by several factors. These factors include a shortage of the materials and/or capacity necessary to produce the vaccine, inefficiencies within the supply chain (i.e. surpluses in some regions and shortages in another), and competition between governments hoping to inoculate their own population before other governments have the opportunity to purchase or manufacture sufficient quantity for their population. The World Health Organization (WHO) has the authority to prequalify vaccines for multi-national distribution. In order to receive this designation a vaccine manufacturer shares with WHO the production process and quality control measures and is also subject to compliance testing and monitoring of complaints. This requires an understanding of the supply chain, international standards, and the regulations drug manufacturers are subject to which will hasten the distribution of a successful vaccine [37, 46].

With the discovery of a vaccine for COVID-19 assumed, manufacturers have begun preparing for the dramatic increase in capacity that will be necessary to meet demand. Tens of companies have already voluntarily outlined the unused capacity within their production lines and have publicly committed to alleviating the supply shortage once a vaccine is found. Though, the continued production of existing vaccines and the ingredients necessary to produce them will take precedence. Thus, a complete picture of the surge capacity of global vaccine manufacturing is unknown as each company will make decisions based on their unique circumstances, which are unknown to the public and can fluctuate. Tackling the COVID-19 production challenge can be achieved through effective communication between manufacturers and the global immunization community. The Developing Countries Vaccine Manufacturers Network (DCVMN) is a public-health body that facilitates the communication necessary to minimize the time from vaccine discovery to deployment within populations . This reduction is achieved by the collection and sharing of information as to quality control and production capacity, providing regulatory oversight, mobilizing international cooperation, and improving immunization standards within member countries [37]. However, the global manufacturing capacity will never reach a point of being able to produce enough vaccine for everybody around the world at once; strategic immunization that improves epidemic outcomes is needed.

Upon the delivery of a new vaccine, a country must decide how to allocate it among the members of their population [1, 36]. This could be simple such as the order in which individuals request to receive the vaccine or complex, by analyzing and extrapolating from the demographic, health, sex, and age information of their population. Regardless, a selection mechanism is required. This leads to the production of a vaccination strategy. In [44, 50] four strategies were considered. These included a random selection of individuals to be vaccinated, and three other strategies based on the structure of the personal contact network, which is known ahead of time. These were done using an improved version of the compartmental Susceptible-Infected-Removed (SIR) model, modified to include a variable rate of infection using the personal contact network of the population. This model uses differential equations to simulate an epidemic and provides only the number of susceptible, infected, or removed individuals at each time step. The next section will outline the compartmental SIR model. Whereas, Case Study 7 features work using a network-based SIR model in which infections can move through the personal contact network only along network edges. The Case Study investigates three potential vaccination strategies for deployment of a new vaccine.

2.3 Epidemic Models

2.3.1 Compartmental Differential Equation Model

The compartmental differential equation model of infection was first outlined in 1927 and is the starting point for most epidemic modelling that is done today [30,33]. The SIR model assumes a well-mixed population in which anybody can infect anyone else within the population. In this model the population is divided into three mutually exclusive groups: those still able to be infected by the epidemic are *susceptible*, those that currently have the epidemic are *infected*, and those that were previously infected are *removed* (due to immunity or death). Using differential equations the number of people within each group is updated every time step. This permits the modelling of simple epidemics on homogeneous populations. The variables of the differential equation can be updated as an epidemic spreads through a population; allowing for the model's projections to be updated and new epidemic curves to be generated.

2.3.2 Network-Based SIR(S) Model

The work featured within the Case Studies uses the compartmental model of infection [30, 33], but substitutes the differential equations out for personal contact networks. These networks limit the spread of a virus along the edges of the network. This permits the researcher to determine how an epidemic spreads through a population from person to person or through communities within the overall population. Additionally, this allows for assigning demographic and health information to individuals within the population to make decisions about individual infections based on an individual's characteristics, rather than relying only on population level statistics. An epidemic begins by choosing one individual within the population to be infected, and then the epidemic is permitted to spread along edges of the network. An individual has a probability α of being infected by each adjacent infected population member, calculated independently. The epidemic disease lasts a single time step within an infected individual after which they are no longer able to spread the epidemic and now belong to the *removed* group. In contrast, the Susceptible-Infected-Removed-Susceptible (SIRS) model of infection adds the ability for an individual to lose immunity, becoming susceptible again after being removed for a number of time steps. The Case Studies use both the SIR and SIRS epidemic models.

2.3.3 Communities

In Chapter 6 we introduce the concept of sub-communities known as districts. Therefore, the overall community or population is itself comprised of a fixed number k of smaller districts. To simplify analysis, all districts have the same size and same structure. Essentially, a personal contact network (graph) of the required district size is evolved and then k copies are used to construct the overall community. Each of the k districts is connected to each of the other districts. A district yet to experience the epidemic has a probability α' that one of its members will become infected by a member of another district that is infected, calculated independently for each neighbouring infected district. Figure 2.1 demonstrates how a community of districts is created using a personal contact network. For simplicity, the first individual infected within a district is the individual represented by the vertex with the lowest index within the district, and is called *patient zero*.

Prior to initiating an epidemic the size and number of districts, k, is used to generate the community; recall that the overall community is modelled as a set of k identical districts which are fully connected to one another as shown in Figure 2.1. The connections between the districts provide the ability for an epidemic to spread between districts. To commence each epidemic, the vertex with the lowest index, *patient zero*, from the district with the lowest index, *district zero*, is marked infected

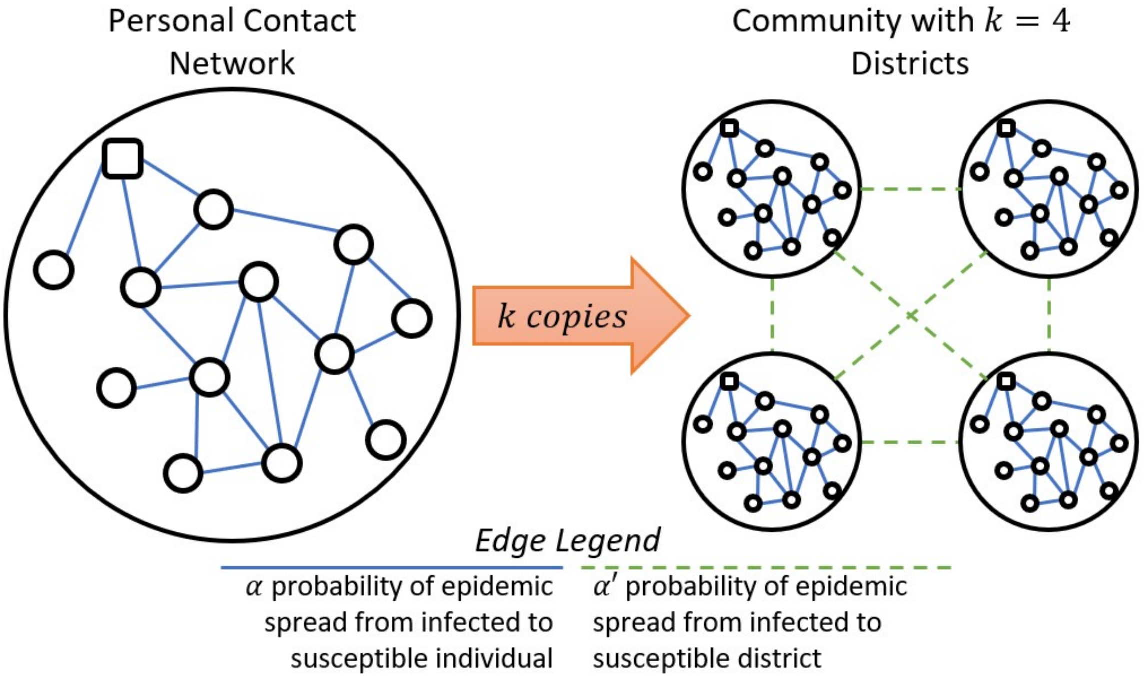

Personal Contact Network
Community with $k = 4$ Districts
k copies
Edge Legend
α probability of epidemic spread from infected to susceptible individual
α' probability of epidemic spread from infected to susceptible district

Figure 2.1: A community is comprised of k identical personal contact networks which are completely connected to one another. The rectangular vertex represents the vertex with the lowest index within the district (patient zero).

and the epidemic is permitted to spread along edges from vertex to vertex within a district. This is known as a *within-district infection*. Within-district infections have probability α of spreading from infected to susceptible individuals within the same district via edges in the graph; each of these probabilities is calculated independently.

The epidemic can also spread from any district with infected individuals to any district yet to experience any infections. This is a *between-district infection* and has probability α' of occurring at each time step. When a new district is first infected, it is again *patient zero* who is the first individual infected within that district.

2.3.4 Simulations

An epidemic, as realized in the Case Studies, begins by choosing one individual within a community to be infected. From there, the epidemic is permitted to spread along edges of the network. The epidemic disease lasts a single time step within an infected individual; after this time they are no longer able to spread the epidemic and now belong to the removed group.

In earlier research, networks that yielded a long epidemic duration were found to be ”banana” shaped, with patient zero at the end of the banana, and the epidemic proceeding down the length of the banana – the thickness of the banana is just enough to prevent the epidemic from burning out early [8]. If the probability of epidemic spread is high, then longer, thinner bananas are generated.

In accordance with the SIR model of infection, all individuals in the population are initially set to the *susceptible* state except for one individual (*patient zero*) who is chosen to be infected with the disease. The status of members of the population is represented by integers in an array; the first element of the array, vertex zero, is set to be patient zero for all epidemics simulated in the Case Studies. Furthermore, the epidemic is then permitted to spread throughout the personal contact network along edges from infected individuals to those who are susceptible. Every infected member can infect each of their neighbours with a probability of $\alpha = 0.5$, calculated independently. This simplification allows us to analyze the impact of a vaccination strategy on an outbreak; in real life situations this value could vary depending on level and duration of contact between individuals.

2.4 Graph Theory

The personal contact network used in this work is implemented as a combinatorial graph. Individuals are the *vertices* of the graph and the connections between individuals are its *edges*. The terms network and graph are used interchangeably within this thesis. A graph G is defined as a set of edges E and vertices V and is denoted $G(V, E)$. An edge is represented as $\{p, q\}$ in which p and q are vertices from V. Only undirected graphs are used: infection can pass in either direction. A path from vertex p to vertex q on graph G is a sequence of edges from E which connect p and q. The *distance* from p to q is the length of the shortest path which connects p and q.

A *path* from vertex p to vertex q on graph G is a sequence of edges from E which connect p and q. The *distance* from p to q is the length of the shortest path between p and q. A graph is *connected* if there is a path from every vertex to every other vertex [49].

2.4.1 Social Contact Networks

This thesis focuses on the ability to generate personal contact networks, representing physical connections between community members, which satisfy the data about the number of infections per time period or maximize epidemic length. A personal contact network is the foundation of an epidemic model in which an epidemic spreads along the links of the network. The case study in Chapter 8 compares two models of disease spread, in preparation for incorporating an asymptomatic state to match the behavior of SARS-Covid-2. The approach of employing a generative solution to a test problem is known as *graph induction* which has a variety of applications [16, 29, 34]. The representation used within this thesis is known as the Local THADS-N generative representation; the metrics used to evaluate the performance of a network are epidemic *profile matching*, introduced in [11], and *maximizing epidemic duration*.

Chapter 3

Evolutionary Algorithms

Darwin, who founded the idea of natural selection, offers us the foundation for the ideas used in evolutionary computation [15]. He observed that random minor alterations to a species, across time, permitted some members of the population to outlive others eventually causing more members of the species to acquire said alteration. Those who had the modification would have more offspring, live longer, and/or be more fit for the environment which granted those genes a greater chance of reproductive success. Across successive generations of the species those without the modification would die off, while those with the modification would thrive. This process is constantly occurring in all biological systems and is the basis for the diversity that exists in nature. The successive advances in living organisms are stored in the DNA that codes the cells and proteins that comprise the organisms [25].

The advancement of living things is realized by a combination of selection bias by the organism, reproduction, and random mutation within the genome. Animals often select mates who demonstrate an ability to raise, support, and protect a family unit; this represents the selection mechanism in animals with these judgements acting as a proxy for biological superiority. When two animals or plants procreate their children share a combination of their parents' DNA, creating a unique genetic code. Though this re-combination of genes is random in nature, with no guarantee that any supposed improvement is passed along to offspring. In fact, the same advancement can evolve independently; the process of analogous features evolving in two different species with their common ancestor not having that feature is known as convergent evolution. One example of convergent evolution is the development of advanced eyesight in mammals and cephalopods, such as octopus, with the most recent common ancestor having only basic photo-receptive cells. Mutation is a result of errors arising within the DNA of a living organism; these errors produce new DNA sequences which can be passed on to offspring. There is no guarantee that any such sequence is passed on to or expressed in an organism's descendants, although these mutations are necessary for the exploration of new, and potential beneficial, DNA sequences [25].

These concepts are the inspiration for evolutionary algorithms which aim to exert the same evolutionary pressure on potential solutions to a test problem that evolution has facilitated on living things, that is, the evolution of an organism's genome to allow successive generations to survive and thrive within their environment to a higher degree than that of their ancestors. This gradual improvement, over millions of years for living organisms, can be simulated on a test problem in a fraction of the time using modern processors. In order to simulate this evolution a researcher must decide on how to articulate the problem, or more accurately the potential solutions to the problem, in a manner which allows for evolution. This includes how candidate solutions are *represented*, *initialized*, *selected* for reproduction, undergo *reproduction*, experience *mutation*, and are *evaluated* in comparison to one another [41]. Each of these will be explained below using the problem of generating a linear line of best fit, with function $y = mx + b$, for a scatter plot populated with points. An example of a line of best fit on a scatter plot can be found in Figure 3.1. It is important to note that methods exist to generate a line of best fit without the use of evolutionary computation, thus this problem is chosen for demonstration purposes only [42].

3.1 Representation

The first hurdle to using evolutionary algorithms on a test problem is choosing a suitable representation for potential solutions to the problem. Like DNA in living things, this representation must be robust enough to represent any and all potential solutions to the problem at hand. Typically, a string of letters and/or numbers are used to represent a solution within the algorithm with rules to convert the string representation to a solution to the real-world problem being solved [41]. When it comes to generating a line of best fit a suitable representation would be two real values, which represent the m and b values of the function describing the line of best fit. This has the ability to represent any straight line, except for a vertical asymptote, a trivial solution that will only arise when the scatter plot has all the points sharing the same value for x. Thus, each potential solution, or chromosome, is comprised of two real numbers.

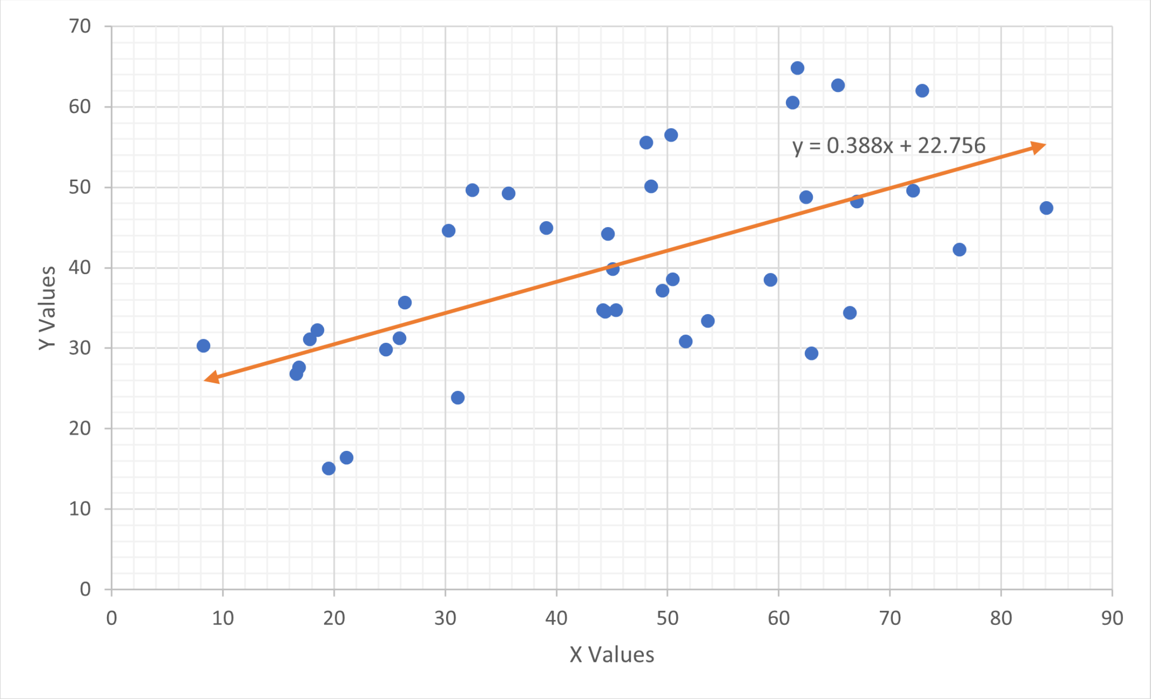


Figure 3.1: A scatter plot with 40 points in blue and an appropriate line of best fit in orange. The equation of the line of best fit is also provided.

3.2 Initialization

Once a representation of the problem has been established a population of random solutions is generated, known as the initial population. The number of chromosomes in this population must be chosen and/or determined during parameter tuning. Regardless, each candidate solution is typically initialized in a random fashion in order to ensure the population as a whole features possible solutions that sufficiently encompass the solution space. However, In the event that the researcher is aware of domain information regarding acceptable solutions they can put appropriate bounds on the randomness of initialization. Furthermore, certain problems may have known, close to optimal, solutions to the problem which can be used as initial solutions themselves or they can be modified in a random fashion to generate the initial population [41]. For example, when considering generating the line of best fit for the points in Figure 3.1 choosing only positive values for m would be beneficial as there is an obvious positive slope to the points. Additionally, limiting the potential values of b to the range $[-70, 70]$, with 70 being the largest value on the Y-axis, would also be logical. Though if these observations are omitted then the values m and b of each chromosome could be set to a random number in the range

$[FLOAT.MINIMUM, FLOAT.MAXIMUM]$, determined by the chosen coding language.

3.3 Fitness Evaluation

After initializing the starting population of solutions to the problem the researcher must determine some way of ranking the solutions in order to compare them to one another. Usually, this is accomplished by representing the quality of a solution with a single real number, permitting a continuous spectrum on which to judge the fitness of a solution in comparison to the other potential solutions. In evolutionary computation, this is termed a *fitness function*; more generally, in optimization problems it is the *objective function* that is to be maximized or minimized. In this work the fitness and objective function are synonymous. There are problems that may require a multi-objective fitness function where a solution is judged on multiple, conflicting or unrelated criteria. In this case, multiple real numbers would be used to evaluate a solution, with each based on a characteristic that an optimal solution would exhibit. For example, an optimal financial portfolio would maximize the expected return while minimizing the potential risk to a client [41]. When considering generating a line of best fit a fitness function for a candidate solution, in the form $y' = mx + b$, could be the sum of the squares $(y - y')^2$ for each x from each point (x, y) in the scatter plot. In this case, the optimal solution would the chromosome with a sum, or candidate solution fitness, of as close to zero as possible.

3.4 Selection

In order for the population of candidate solutions to converge towards an optimal or nearly-optimal solution across simulated generations a mechanism for determining who within the population is granted reproductive rights is necessary. Two common selection mechanisms are: roulette wheel selection whereby each member of the population is granted a likelihood of being chosen proportional to the quality of the solution and tournament selection in which a random subset of the population is selected and the two solutions with best fitness in the tournament are chosen to reproduce. In addition to the selection of parents, a researcher must determine how to manage the new children and the rest of the population not chosen for reproduction. A *steady state* evolutionary algorithm is one that retains a majority of the population, unchanged, between generations; with the children replacing members of the population with the worst fitness. Whereas, a *generational* evolutionary algorithm replaces all or the vast majority of the population with offspring generated using one (asexual reproduction) or two (sexual reproduction) members of the old generation. In this case if some amount of the population is retained between generations this is known as *elitism* as those with the best fitness are the solutions retained [41]. In the case of determining a line of best fit any combination of these selection mechanisms could be chosen.

3.5 Reproduction

After selecting which two solutions are to be used for reproduction a process to create new solutions, the children, using genetic recombination needs to be determined. Similar to biological evolution a child is created using the strings representing the two parent chromosomes. Four common recombination strategies include: single-point crossover, two-point crossover, k-point crossover, and uniform crossover. In order to demonstrate how these strategies make children, consider a hypothetical evolutionary algorithm in which each chromosome is represented as a string of 100 binary digits (i.e. 100 0's or 1's). The two parent strings A and B will be used to generate two child strings c and d. When using single-point crossover a random index, i in the range $[0, 99]$ is chosen which is known as the crossover point. Then the first child string, c, is created by copying the first i digits from A followed by copying the last $100 - i$ digits from B. Similarly, d is generated by copying the first i digits from B followed by copying the last $100 - i$ digits from A. In two point or k-point crossover this idea is extended to include two or k indices; with the genetic material comprising the children being copied from the parents, with the source flipping between parents at each index. Lastly, uniform crossover randomly selects which parent is the source at each digit in the string, with the second child receiving the genetic material not given to the first child. Thus, all the digits which exist in the two parents will be represented in the children regardless of the recombination strategy being implemented. Additionally, a crossover probability must be chosen by the researcher representing the likelihood that two parents will undergo crossover to generate offspring [41].

Switching back to the problem of determining a line of best fit, the crossover strategies described above don't map directly onto this problem. A chromosome in this case is two floating point values, namely m and b from the equation $y = mx + b$. However, we can use a similar idea by calculating the difference between the two parents' m and b values denoted δm and δb. Additionally, let the smaller slope between the parents be m_m and the smaller y-intercept be b_m. The first child, c, would be generated as follows $m = m_m + \frac{1}{3}\delta m$ and $b = b_m + \frac{1}{3}\delta b$. Similarly, the second child would be $m = m_m + \frac{2}{3}\delta m$ and $b = b_m + \frac{2}{3}\delta b$.

3.6 Mutation

Now that new children have been generated a mechanism to simulate mutation must be chosen by a researcher. The mutation mechanism achieves the randomness in nature that allows for new genetic material to generated permitting the evolutionary algorithm to further explore the solution space. Some common mutation operators include: flip bit, which works on binary strings by randomly flipping some number of bits in the chromosome to the opposite value (i.e. 0 becomes 1 or 1 becomes 0); uniform mutation, which replaces a number of values in the chromosome with a randomly generated value between a user-specified upper and lower bound; and Gaussian mutation, which works similarly by adding a random value, from a user-defined normal distribution, to some number of values within the chromosome. Considering the level of randomness present within some of these mutation operators it may be necessary to disallow new chromosome values that fall outside upper and lower bounds decided by the researcher. In addition to choosing a strategy, the probability of mutation to occur as well as which chromosomes undergo mutation must be determined. Typically the mutation probability is low for any particular value comprising a chromosome; one option is to set the probability to $\frac{1}{length of chromosome}$ meaning each chromosome should expect to have one value undergo mutation. Furthermore, mutation could be applied to every chromosome in the population or reserved for the newly generated children [41]. When solving for a line of best fit Gaussian mutation would be an appropriate mutation operator to be applied, probabilistically, to the newly generated children chromosomes.

3.7 Parameter Selection

An important consideration for evolutionary algorithms is an appropriate exploration of the parameter space. Each algorithm, regardless of the decisions made for each component above, involves several probabilities that will have a significant impact on the algorithm's ability to find acceptable solutions. Thus, testing several combinations of probabilities, upper and lower bounds, and choices for operators is necessary to tune the algorithm and achieve acceptable results [41].

3.8 Evolutionary Computation

Evolutionary algorithms represent one of the many concepts which encompass evolutionary computation more generally. Some of these algorithms include: genetic programming which evolves programs or functions typically represented in a tree structure where leaf nodes hold operands and tree nodes hold operators [48]; ant colony optimization which simulates the behaviour of ants using locally available information to each ant and pheromones to signal choices that have benefited other ants in the population [28]; and particle swarm optimization in which each particle maintains a velocity and undergoes an acceleration towards particles with a better fitness [32].

Chapter 4

Methodology

4.1 The Generative Representation

The Local THADS-N representation does not explicitly provide a solution to a problem. Instead it provides a number of edge-editing operations (Toggle, Hop, Add, Delete, Swap, and Null) performed on an initial graph in order to specify a solution to the problem. A generative solution is chosen because it permits the inclusion of domain information, such as a reasonable number of edges, in the initial graph. It also permits evolution of graphs with a simple linear structure, a list of editing commands represented as integers applied to an initial graph. Some existing applications of generative representations include [16, 29, 34] and evidence of the effectiveness of generative solutions is shown in [31]. The operations within this representation dictate changes to the edge space of a graph. The initial graph in which these strings of operations are applied is shown in Fig. 4.1. This graph was chosen because previous research [8, 11] has demonstrated that graphs having vertices with four or five edges are desirable for the test problems. Other graph topologies appear in previous work [3, 6, 8–10, 12].

In [7] the first operation of the representation, edge swap, was introduced as a potential universal operator for graph induction. Edge swap was used because it was able to take an initial personal contact network and change which members of the population come into contact with each other. Additional operations were added to the representation in [3], allowing for the addition and deletion of connections; more recent work has focused on adding local variants of the operations [12, 19, 20]. Local operations are only applied to triples of vertices for which any pair of the vertices are at a maximum distance of two before and after the operation is applied. Previous research has shown that local operations have great potential for achieving the proper

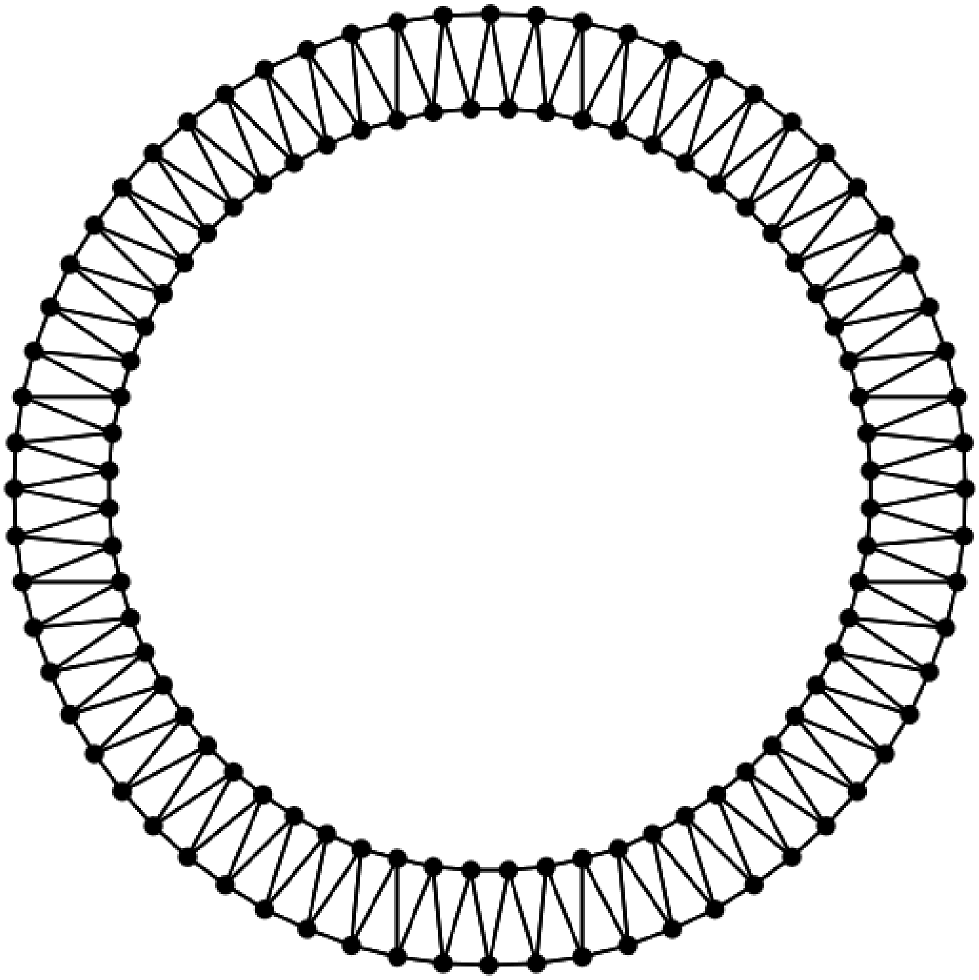


Figure 4.1: Initial graph with 128 vertices on which to apply the string of edge operations. Each vertex has two edges to the two preceding nodes as well as two edges to the two proceeding nodes in the ring.

balance between exploration and exploitation in the evolutionary algorithm.

Given a graph $G(V, E)$ and the vertices p, q, r, and s from the set V the edge operations are defined below and can be visualized in Figure 4.2.

- **Toggle(**p**,** q**):** If edge $\{p, q\}$ is in E then remove $\{p, q\}$ from E, otherwise add $\{p, q\}$ to E.
- **Local Toggle(**p**,** q**,** r**):** If edge $\{p, q\}$ and $\{q, r\}$ are in E then **Toggle**(p, r).
- **Hop(**p**,** q**,** r**):** If edge $\{p, q\}$ and $\{q, r\}$ are in E and edge $\{p, r\}$ is not in E then remove edge $\{p, q\}$ from E and add edge $\{p, r\}$ to E.
- **Add(**p**,** q**):** If $\{p, q\}$ is not in E then add $\{p, q\}$ to E, otherwise do nothing.
- **Local Add(**p**,** q**,** r**):** If edge $\{p, q\}$ and $\{q, r\}$ are in E then **Add**(p, r).
- **Delete(**p**,** q**):** If $\{p, q\}$ is in E then remove $\{p, q\}$ from E, otherwise do nothing.
- **Local Delete(**p**,** q**,** r**):** If edge $\{p, q\}$ and $\{q, r\}$ are in E then **Delete**(p, r).
- **Swap(**p**,** q**,** r**,** s**):** If $\{p, q\}$ and $\{r, s\}$ are the only edges between p, q, r and s then remove $\{p, q\}$ and $\{r, s\}$ from E and add $\{p, s\}$ and $\{q, r\}$ to E.
- **Null():** Do nothing.

4.2 Evolution

A steady state evolutionary algorithm [43] is used to generate the strings of edge operations which correspond to a solution. The variables with respect to system design were determined empirically.

A population of 1000 chromosomes each containing a string of 256 Local THADS-N edge operations is used. When a single string of operations is applied to the 128-vertex graph in Fig. 4.1 a candidate solution to the test problem is produced. The chromosomes are initially generated at random based upon the probabilities provided to each of the operations via the program parameters. From there, the chromosomes undergo 500,000 mating events, with statistical output being recorded every 5,000 mating events for the population in the ED and ES problems. The PM problem uses 40,000 mating events with output every 400 events as fitness calculation is much more intensive. Each mating event consists of a round of tournament selection, crossover and mutation. Tournament selection selects 7 chromosomes at random from the

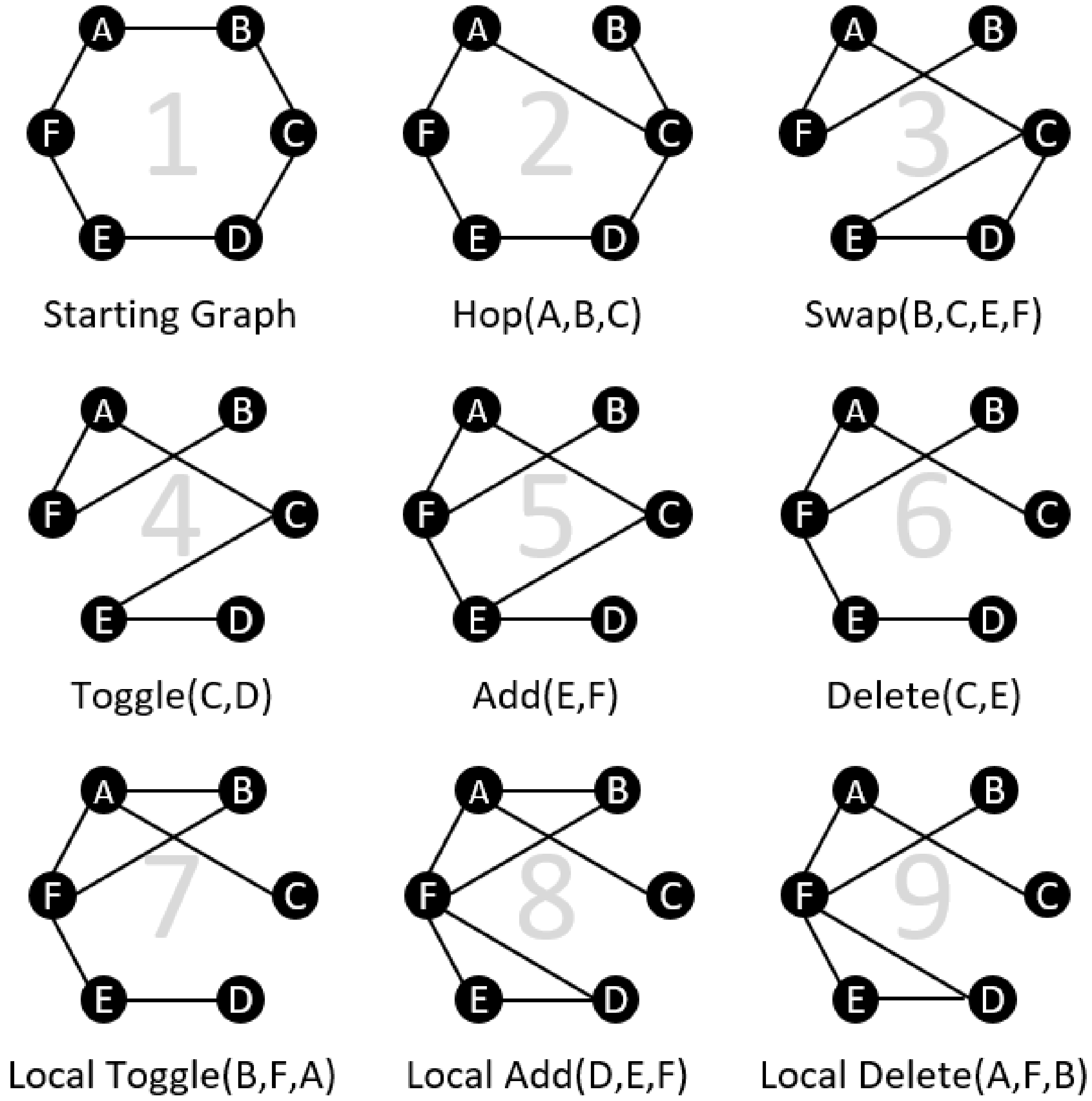


Figure 4.2: Examples of operators included in the Local THADS-N representation. The figure shows the eight of the nine operations being applied sequentially to an initial six cycle graph. The operations are applied in reading order such that the final graph is the result of applying all eight operations. The null operation is omitted as it does not change the graph.

population. The two chromosomes with the worst fitness are replaced with copies of the two chromosomes with the best fitness. Next, the two copies undergo two-point crossover (probability of 1), and mutation occurs on 1-3 of the operations within that chromosome, replacing them with new commands chosen by the same probability distribution discussed above. The choice of 1-3 mutations is randomly determined with each choice being equiprobable. Lastly, the fitness of the children is recalculated. After evolution the candidate solution with the best fitness from the whole population is saved. This process is repeated 30 times for each parameter setting (PS) being tested. In the PM problem this is repeated on each of the nine epidemic profiles in Figure 4.3, 4.4, 4.5, 4.6, and 4.7.

In order to determine which solutions should be favoured for evolution two fitness functions are used. Both simulate epidemics on solutions in the form of personal contact networks. In each epidemic the vertex with the lowest index, *patient zero*, is marked infected and the epidemic is permitted to spread along edges from vertex to vertex. These epidemics have probability $\alpha = 50\%$ of spreading to susceptible individuals via edges in the graph; each of these probabilities is calculated independently. It is important to note that this fitness measure does not indicate the absolute quality of a network. Instead, it measures the relative quality of a network, permitting successive candidate solutions to converge to networks which are more likely to create epidemics satisfying the problem.

4.2.1 Tournament Selection

Each mating event consists of a round of tournament selection, crossover and mutation. Tournament selection selects 7 chromosomes at random from the population, evaluates their fitness and replaces the two chromosomes with the worst fitness by copies of the two chromosomes with the best fitness. These two copies then undergo two-point crossover, and mutation occurs on 1-3 of the operations within that chromosome, replacing them with new commands chosen by the same probability distribution. The choice of 1-3 mutations is randomly determined with each choice being equiprobable. Finally, fitness is recalculated for the children.

4.2.2 Skeptical Tournament Selection

The epidemic duration and spread problems calculate fitness by simulating a single epidemic on a candidate solution. The length of the epidemic is measured as the number of time steps until there are zero infected individuals in the (overall) community.

This single sample epidemic is used to estimate the fitness of a solution, implying that a mediocre community may attain a fitness value for this single epidemic that is far greater than its mean fitness across epidemics. To circumvent a scenario in which a solution is provided undeserved reproductive rights *skeptical tournament selection* is used [45]. This modifies traditional tournament selection by recalculating the fitness of the parents after a mating event. This reduces the probability of re-selecting a solution which once had a sample fitness far above its average fitness, although this situation will still occur. The use of skeptical tournament selection has been shown to yield better solutions for the epidemic duration problem [45], hence its use. Additionally, this process favours graphs with less variance as solutions which are chosen for reproduction with increased frequency also have their fitness recalculated more frequently.

4.3 Fitness Evaluation

It is important to note that this fitness measure does not indicate the absolute quality of a network. Instead, it measures the relative quality of a network, permitting successive candidate solutions to converge to networks which are more likely to create epidemics satisfying the problem. All the fitness functions used feature a many-to-many relationship in which the value achieved by the fitness function can potentially map to many personal contact networks. This is because the fitness is not determined by the edge space of the networks, but rather by unleashing epidemics on the networks. In this work the fitness function and objective function are synonymous.

4.3.1 Profile Matching Fitness

Introduced in [11], epidemic Profile Matching (PM) begins with defined epidemic behavior on a human population to determine if networks likely to permit similar behavior, can be generated. An epidemic profile is specified by the number of individuals infected at each time step of an epidemic simulation. There is no evidence to suggest that a particular network is ideal for any given epidemic profile, therefore the goal of the epidemic profile matching problem is to find networks likely to generate behavior resembling the profile. The nine profiles used to test this problem, from [11], are shown in Figure 4.3, 4.4, 4.5, 4.6, and 4.7. These profiles were chosen as they provide a broad range of potential epidemic behaviour and to be able to compare the fitness with previous research.

The PM problem uses the fitness function from [45] which determines a solution's fitness by simulating 50 epidemics. By comparing the number of infected individuals at each time step of each epidemic with the expected number of infected individuals the sum squared error (SSE) of a solution is calculated. The 50 SSE measurements are then sorted in increasing order $E_1 \leq E_2 \leq \ldots \leq E_n$, which is used to determine the fitness of a graph G in the form of a linearly weighted sum of the measurements according to $\mathit{fit}(G) = \sum_{i=1}^{n} \frac{E_i}{i}$.

The SSEs are sorted, allowing for the fitness function to be most impacted by simulated epidemics which most accurately resemble the known epidemic profile being considered. In order to provide a fitness value for a network which can be compared to other networks the fitness is calculated, without weighting, after execution.

A PM fitness of 0.0 would mean that all the simulated epidemics perfectly recreated the profile in question. However, this outcome is highly unlikely given the stochastic nature present when it comes to an epidemics spreading between individuals within a population. The largest (and worst) fitness possible would depend on the profile under consideration.

A fitness plot of the mean fitness of the population of solutions using parameters setting 1 and profile 1 from [21] is included in Figure 4.8. It can be seen that the population has more or less converged around generation 35 000 with fitness 11.2, from an initial mean fitness of 23.0. Convergence occurs when the evolutionary algorithm only achieves marginal improvements if it is left running.

4.3.2 Epidemic Duration Fitness

The length of an epidemic is the number of time steps until there are zero infected individuals within a population. The Epidemic Duration (ED) problem [8] seeks to find graphs which promote longer-lasting epidemics. Originally, this was determined by averaging the length of 50 simulated epidemics on a candidate graph [6, 8]. In contrast, this fitness function uses one sample epidemic to evaluate the fitness of a solution combined with a selection method known as *skeptical tournament selection*, detailed in Section 4.2.2, from [45].

The best fitness possible would be equal to the number of nodes in the personal contact network being tested, with this representing the virus infecting a single individual at each time step. Conversely, the worst fitness possible would be 2 whereby patient zero infects every other member of the population in one time step. Both of these extremes are highly unlikely given the necessary edge space requirements.

4.3.3 Epidemic Spread Fitness

The *spread* of an epidemic is the total number of individuals infected by the epidemic over its entire course, i.e. from the time at which it is unleashed up until the time at which there are zero infected individuals within the population. As with epidemic length, this is applied both to static graphs and to graphs that evolve in reaction to the vaccination strategy. Evaluation of epidemic spread is a simple sum of the number of newly infected individuals during each time step. Additionally, in order to prevent evolution from coalescing around networks with ever-increasing edge counts the fitness function evaluates to zero whenever the total number of edges is greater than five times the number of vertices. Skeptical tournament selection is used from [45].

The potential minimum for this fitness function is 1 in which patient zero does not infect anybody else whereas the maximum fitness possible would be the number of nodes within the graph, meaning every individual was infected over the course of the epidemic.

4.4 Vaccination Strategies

One of the goals of this study is to evaluate different vaccination strategies with respect to their effect upon (a) epidemic length and (b) epidemic spread. Four strategies are applied to each of these, as follows:

- **No vaccine:** No individuals are vaccinated at any point.
- **Random:** At each time step, one individual from the *susceptible* category is selected for vaccination. Selection is performed uniformly at random.
- **High degree:** At each time step, one individual from the *susceptible* category is selected for vaccination. Selection is performed uniformly at random amongst all of the individuals (vertices) that have the highest degree of all nodes in the graph.
- **Ring:** At each time step, one individual from the *susceptible* category is selected for vaccination. Selection is performed uniformly at random from amongst all of the individuals (vertices) that are neighbours of infected individuals.

As stated earlier, according to our model infection can pass from one individual to a neighbouring individual within one time step. The length of time before a vaccine becomes effective can vary from one vaccine to another. In the current study, the vaccine becomes effective and the individual is added to the *removed* category immediately upon being selected to receive the vaccine according to the current vaccination strategy.

4.5 Point Packing Parameter Selection

A notable concern when implementing evolutionary algorithms is how to appropriately set the numerous parameters fundamental to this type of problem. A popular method to determine suitable parameters is for a researcher to perform pragmatic preliminary experimentation with various PSs. Alternatively, parameters can be chosen based upon appropriate ranges, altered one at a time to determine that parameter's impact on the ability of the algorithm to find optimal solutions. A full factorial exploration of the parameter space is also an option, as evolutionary algorithms commonly interact in a non-linear manner. The representation used in this research has nine parameters, namely the probabilities associated with each of the Local THADS-N editing operations. A full factorial exploration would grow as the eighth power of the sampling density as the nine probabilities must sum to one, removing one of the degrees of freedom. Anything less than a full factorial exploration allows for an optimal parameter setting to never be discovered.

In order to overcome this problem *point packing* is used. Both point packing [5] and the full factorial approach result in a set of points throughout the parameter space, where each point is a PS. This method of parameter selection will result in far fewer points than the full factorial approach. The point packing also allows the researcher to set the minimum spacing between points in the parameter space which determines the number of PSs to be tested. Obviously, the number of PSs tested correlates with the degree to which the parameter space is explored but with a greater cost to run more PSs. In determining a method to select PSs point packing was the most similar to that of a full factorial exploration, but the full factorial's costs grow much faster than those of point packing. Point packing includes many of the same benefits of a full factorial design, namely: thorough exploration of the parameter space and objective design removing any bias the researcher may have. Achieving this level of exploration with far fewer points than a fixed grid.

In this study the chromosome length was set to 256 for all experiments as this was the only parameter that was a large integer; whereas the probabilities associated with the nine operations are decimal values less than one which when summed add to one. These values were set using an evolutionary algorithm that uses point packing.

Algorithm 1. Conway's Lexicode Algorithm

Input: *a set $\mathcal{S}$ of points in some order.*
a minimum distance δ.
Output: *a subset $\mathcal{T}$ of $\mathcal{S}$ with minimum distance δ.*

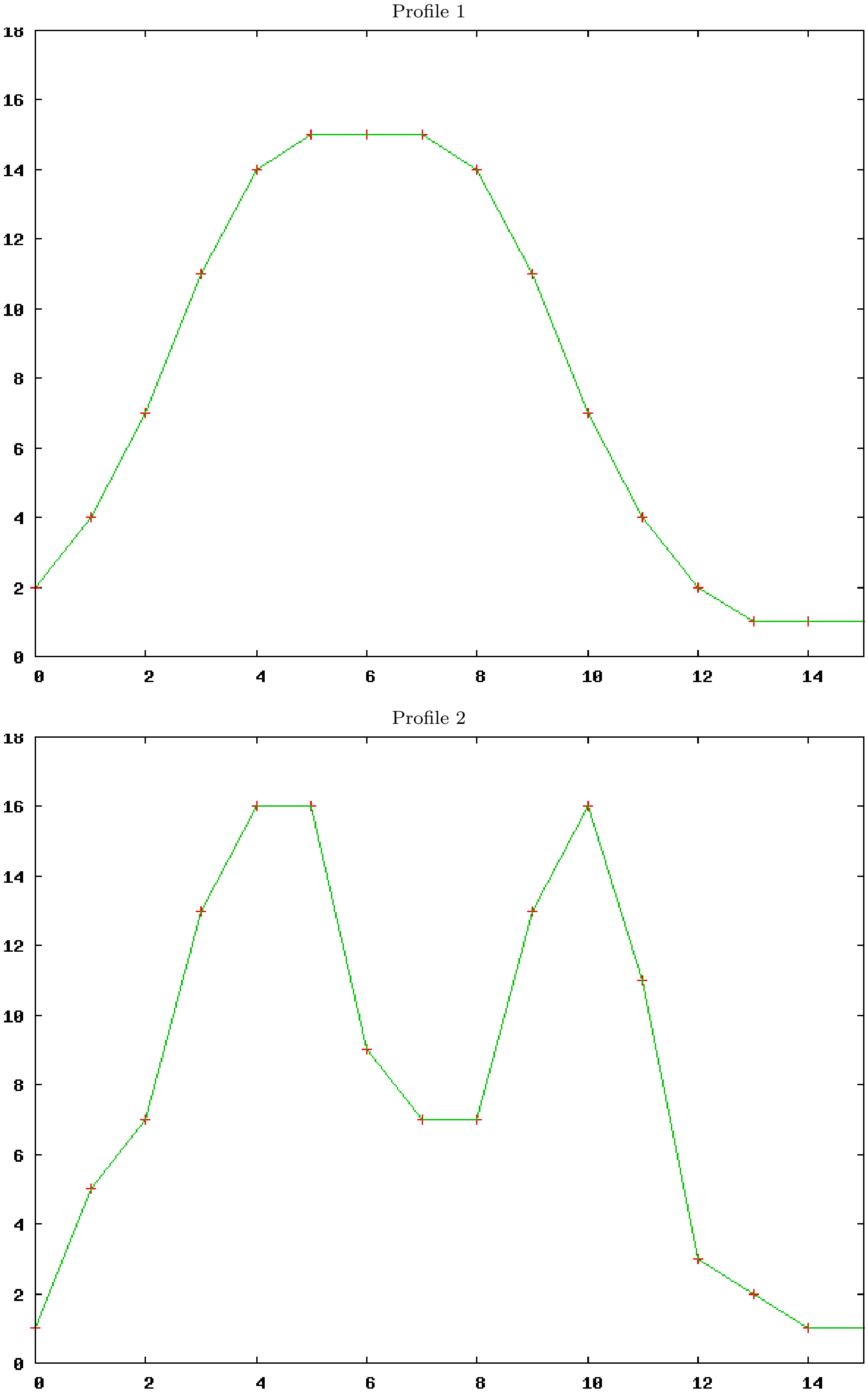
Details:

```
Initialize $\mathcal{T}$ to be empty.
Traversing $\mathcal{S}$ in order,
Add a point from $\mathcal{S}$ to $\mathcal{T}$ if its distance
    from the current members of $\mathcal{T}$ is at least $\delta$.
Return($\mathcal{T}$)
```

The point packing algorithm uses *Conway's Lexicode Algorithm*, shown in Algorithm 1, to generate the initial population of PSs and as the variation operator used in the evolutionary algorithm; see further details in [5]. To run this algorithm, a researcher must choose a value for the minimum allowable distance between PSs.

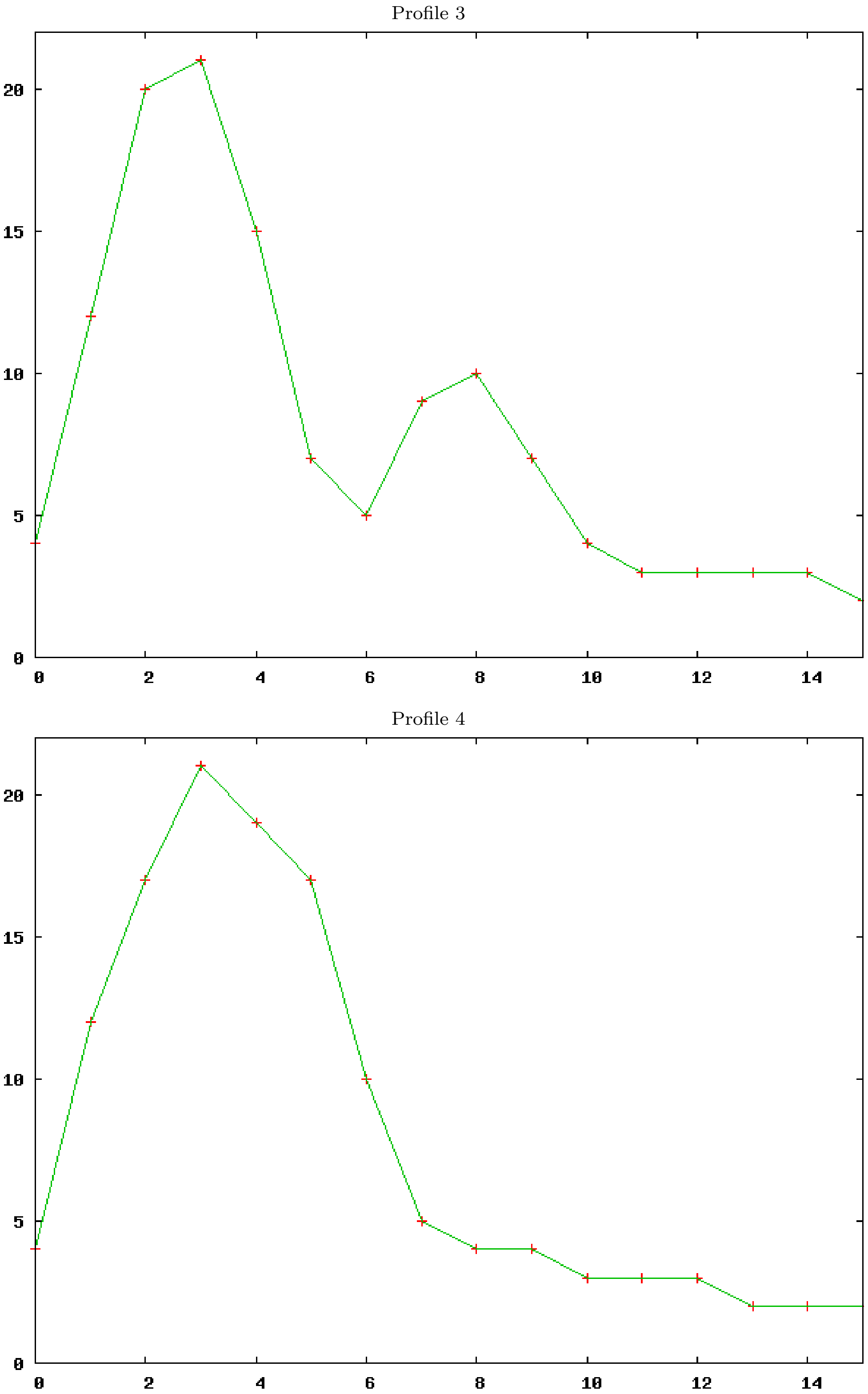
Evolutionary algorithms use crossover to capitalize on solutions that have the best fitness while using mutation to introduce enough randomness to avoid locally maximal solutions. In contrast, the evolutionary algorithm used to achieve a point packing with the most points uses one variation operation: Conway Crossover Operator (CCO). This operator first selects two collections of PSs and combines these sets into one set. From there, twenty new PSs are generated randomly and they are added to the combined set. This set is then shuffled, and a new collection of PSs obeying the minimum distance between points is generated using Algorithm 1. This process accomplishes both crossover, by combining two collections, and mutation, by introducing randomly generated PSs.

The algorithm that evolves point packings of PSs uses a population of 4000 collections of PSs. Each mating event randomly chooses three members of the population and the two with the largest collection of points are subjected to CCO and the newly generated collection replaces the collection with the least number of points. Thirty runs are conducted, each comprised of 1,000,000 mating events. The run that produces a collection with the most PSs is chosen for the experiment because if there are more points obeying the minimum distance in a set space then the space is the best explored using that largest collection of points.


Profile 1

Profile 2

Figure 4.3: Epidemic profiles 1 and 2 representing time step vs. number of infected individuals during that time step.



Profile 3

Profile 4

Figure 4.4: Epidemic profiles 3 and 4 representing time step vs. number of infected individuals during that time step.

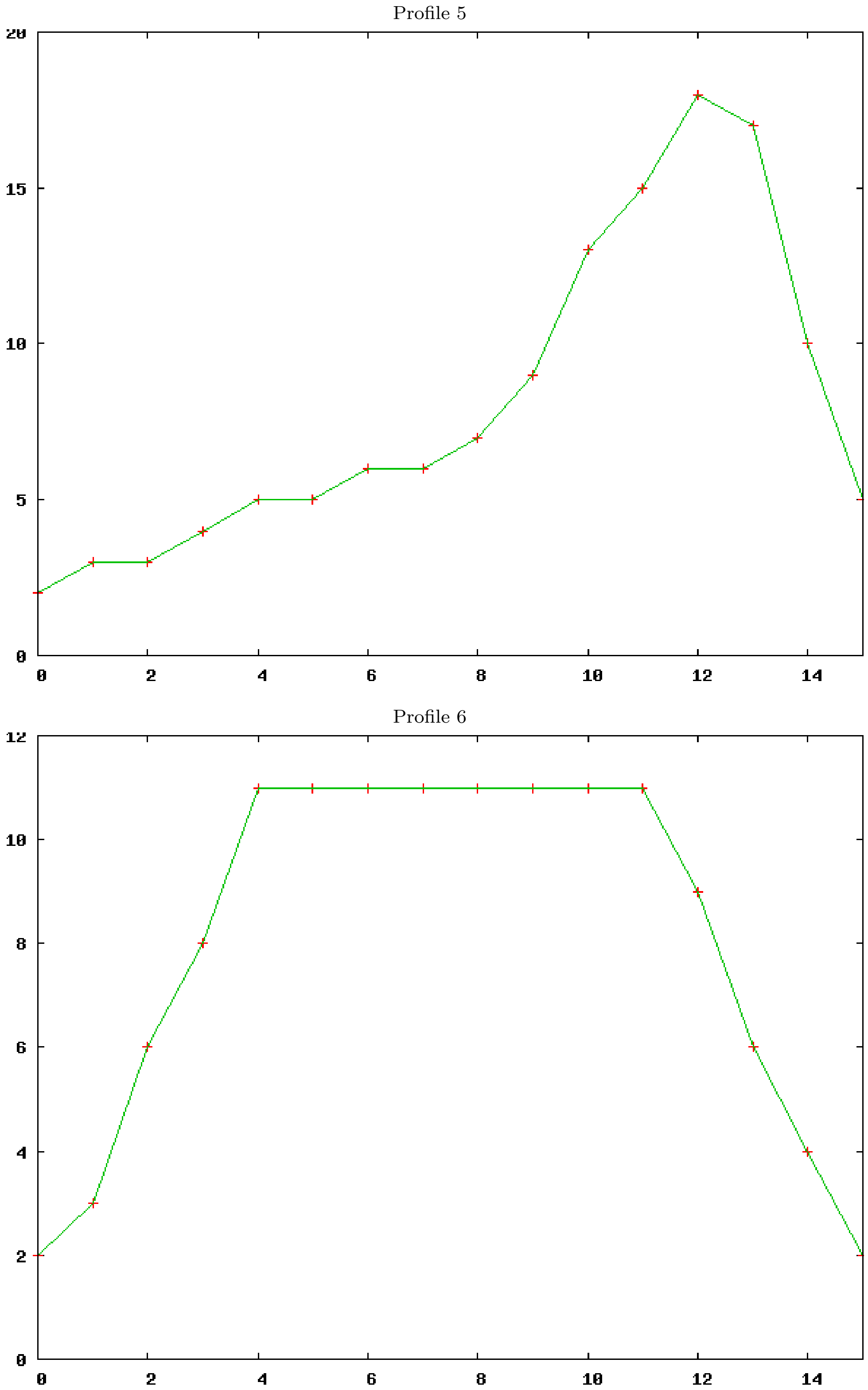


Figure 4.5: Epidemic profiles 5 and 6 representing time step vs. number of infected individuals during that time step.

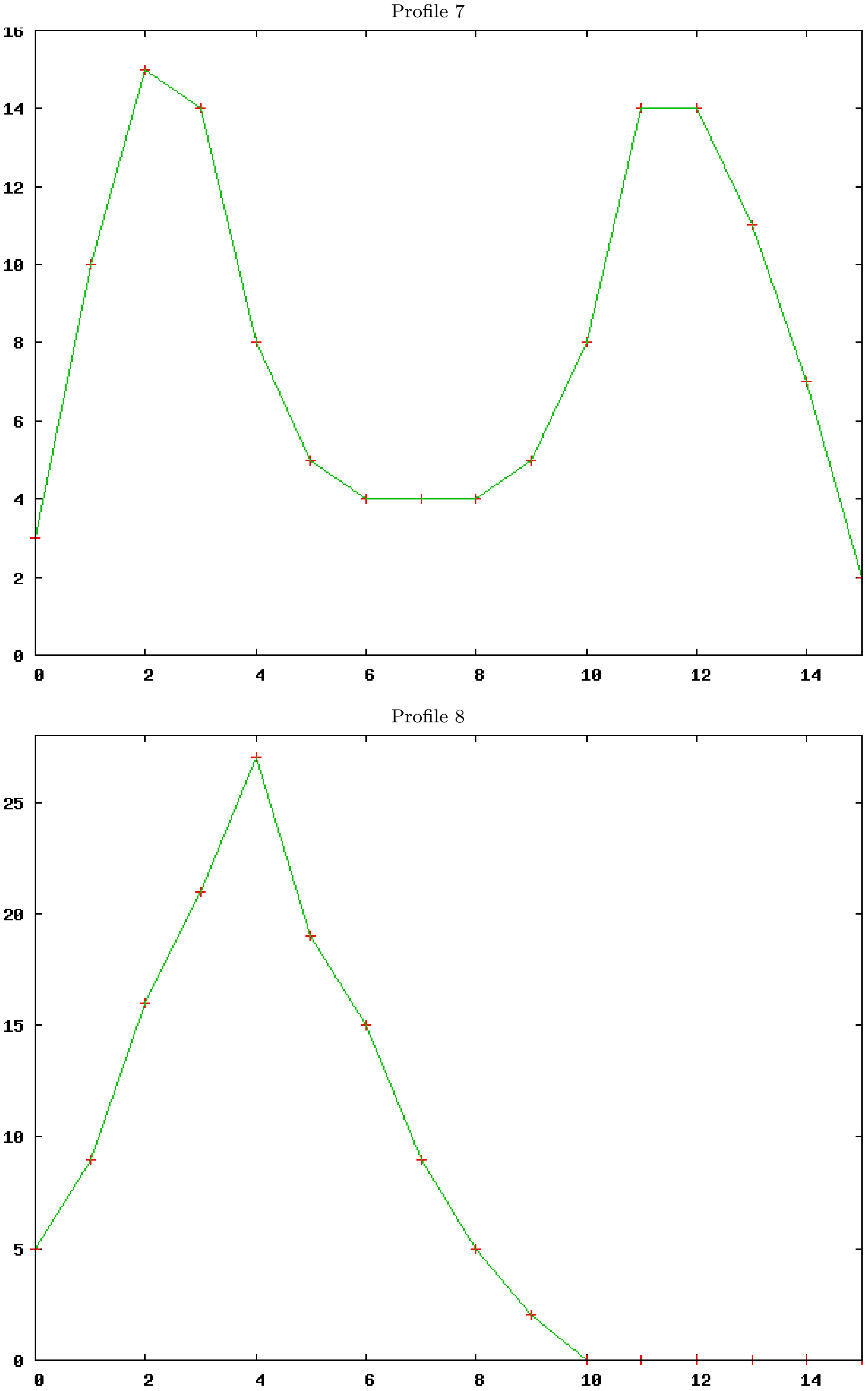


Figure 4.6: Epidemic profiles 7 and 8 representing time step vs. number of infected individuals during that time step.

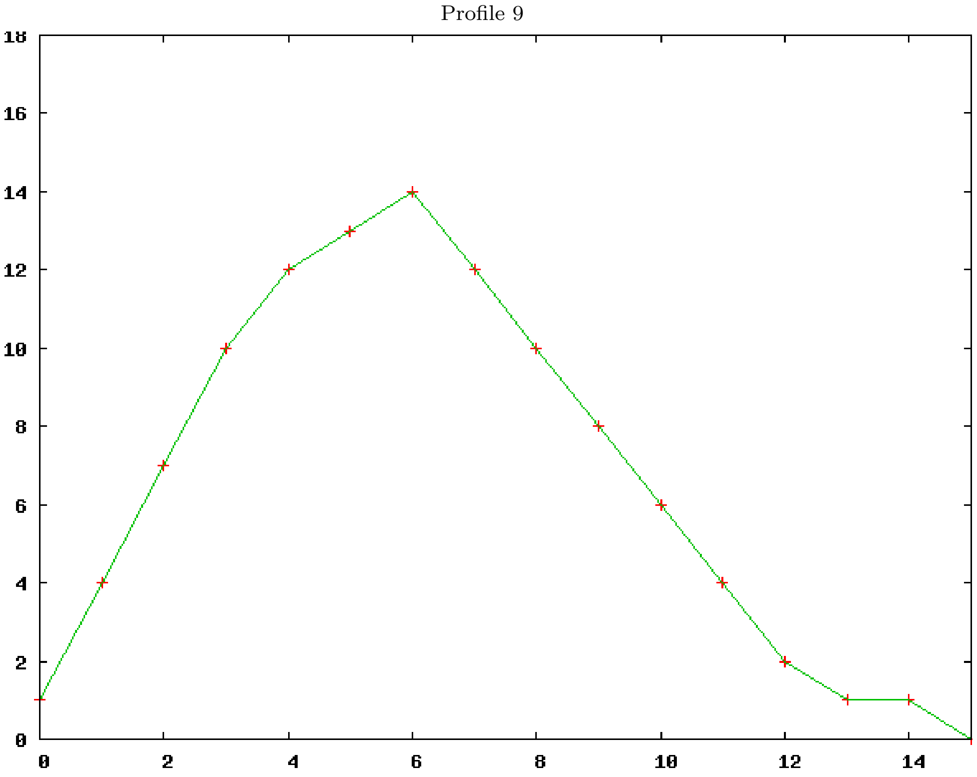


Figure 4.7: Epidemic profile 9 representing time step vs. number of infected individuals during that time step.

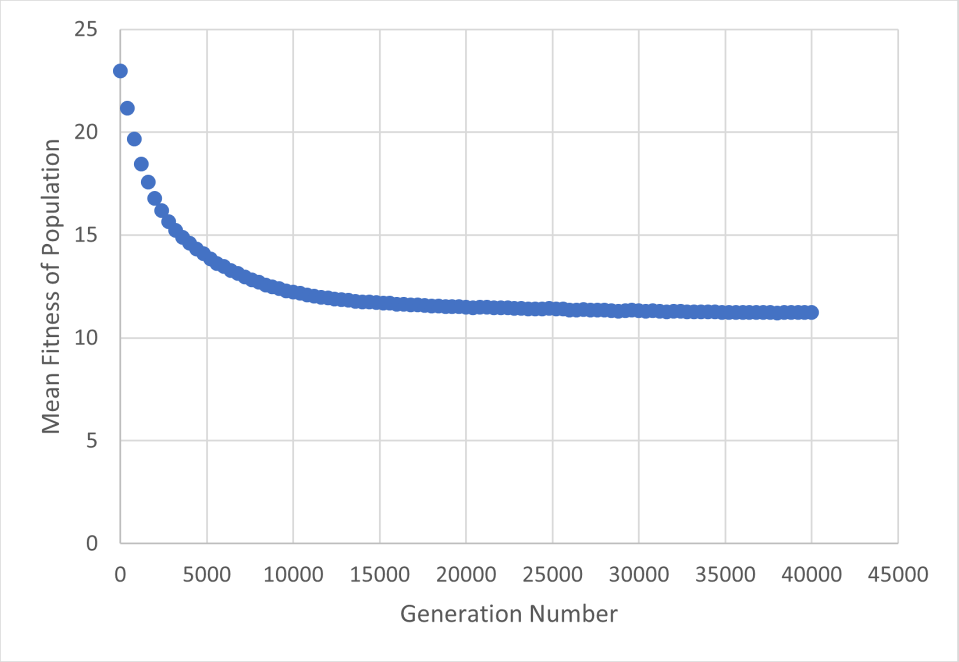


Figure 4.8: The mean fitness of the population of solutions across 40 000 generations using parameter setting 1 and profile 1 from [21]

Chapter 5

Case Study 1

Representation for Evolution of Epidemic Models

This case study was originally published in CEC 2019 conference proceedings [21]. It explores the creation of a representation capable of generating personal contact networks that are most likely to exhibit specific epidemic behavior. This is difficult due to the inherit volatility of an epidemic and the numerous parameters accompanying the problem. To surpass these hurdles, evolutionary algorithms are used to create a generative solution which generates personal contact networks, modeling human populations, to satisfy the epidemic duration and epidemic profile matching problems. This representation is entitled the Local THADS-N representation. Two new operators are added to the original THADS-N system, and tested with a traditional parameter sweep and a parameter selection method known as point packing on nine epidemic profiles. Additionally, a new epidemic model is implemented in order to allow for lost immunity within a population.

5.1 Introduction

The representation used within this paper is known as the Local THADS-N generative representation; the test problems explored here are maximizing *epidemic duration* [8] and epidemic *profile matching* [11].

5.2 The Local THADS-N Representation

5.2.1 Edge Operations from Previous Work

Given a graph $G(V, E)$ and the vertices p, q, r, and s from the set V the existing operations are defined below.

- **Toggle(**p**,** q**):** If edge $\{p, q\}$ is in E then remove $\{p, q\}$ from E, otherwise add $\{p, q\}$ to E.
- **Local Toggle(**p**,** q**,** r**):** If edge $\{p, q\}$ and $\{q, r\}$ are in E then **Toggle**(p, r).
- **Hop(**p**,** q**,** r**):** If edge $\{p, q\}$ and $\{q, r\}$ are in E and edge $\{p, r\}$ is not in E then remove edge $\{p, q\}$ from E and add edge $\{p, r\}$ to E.
- **Add(**p**,** q**):** If $\{p, q\}$ is not in E then add $\{p, q\}$ to E, otherwise do nothing.
- **Delete(**p**,** q**):** If $\{p, q\}$ is in E then remove $\{p, q\}$ from E, otherwise do nothing.
- **Swap(**p**,** q**,** r**,** s**):** If $\{p, q\}$ and $\{r, s\}$ are the only edges between p, q, r and s then remove $\{p, q\}$ and $\{r, s\}$ from E and add $\{p, s\}$ and $\{q, r\}$ to E.
- **Null():** Do nothing.

5.2.2 New Edge Operations to be Tested

Given a graph $G(V, E)$ and vertices p, q, and r from V the new operations examined in the current study are as follows:

- **Local Add(**p**,** q**,** r**):** If edge $\{p, q\}$ and $\{q, r\}$ are in E then **Add**(p, r).
- **Local Delete(**p**,** q**,** r**):** If edge $\{p, q\}$ and $\{q, r\}$ are in E then **Delete**(p, r).

The inclusion of these operations finalizes the study of local variants of operations, and completes the list of all operations in the Local THADS-N representation used in this paper.

5.3 Experimental Design

The specifications for the four experiments in this study are summarized in Table 5.1.

Table 5.1: The specifications for the four experiments included within this paper.

Exp.	Model	Fitness Function	Tournament Selection	Parameter Selection	Parameter Sets	Testing
A	SIR	Profile Matching	Traditional Size 7	Parameter Sweep	26	New Local Add Edge Operation
B	SIR	Profile Matching	Traditional Size 7	Parameter Sweep	26	New Local Delete Edge Operation
C	SIR	Profile Matching	Traditional Size 7	Point Packing	90	Point Packing Parameter Selection
D	SIRS	Epidemic Length	Skeptical Size 7	Point Packing	29	SIRS Model Compared to SIR

5.3.1 Exp. A & B: Local Add and Delete on the PM Problem

In order to determine whether the addition of each of the new operators is beneficial to the representation for the PM problem a traditional parameter sweep was performed. In [45] it is demonstrated that swap allowed for better performance than hop for the PM problem. Additionally, [20] establishes that, for most profiles, the best fitness was accomplished when the probability of toggle and local toggle are approximately equal. Therefore, the percentage for swap is zero for both parameter sweeps. The proportion for local add is also set to zero for the local delete parameter sweep, and vice-versa. The PSs for Exp. **A**, looking at local add, are in Table 5.2. The local delete parameter sweep, for Exp. **B**, follows the same pattern as local add, although the table is omitted due to page limits.

In both sweeps the first PS sets the proportions for original add or delete and its local variant to zero while the other operations, except swap, are given equal proportions. This establishes a baseline for how the evolutionary algorithm performs without any add or delete. The rest of the PSs are divided into five sets of five PSs each. The PS of the first set for both the add and delete parameter sweeps assigns 0.1 to add or delete and zero to their new variants. The remaining 0.9 is evenly divided by the other operations as in PS 1. Each successive PS within the set transfers a portion of the probability of the original operation to its local variant. The remaining sets perform the same operation with a greater proportion being shared by the two operations being compared. This allows for the performance of the original operation to be directly compared to its local variant.

5.3.2 Exp. C: Point Packing on the PM Problem

The point packing is being used as a tool to generate PSs for use in Exp. **C**. The point packing algorithm uses *Conway's Lexicode Algorithm*, shown in Algorithm 1,

Table 5.2: Traditional parameter sweep for Exp. **A** on the PM problem. The same pattern is repeated for Exp. **B** on local delete.

PS	Togg.	Hop	Add	Del.	Swap	L-Togg.	L-Add	L-Del.	Null
1	0.2000	0.2000	0.0000	0.2000	0.0000	0.2000	0.0000	0.0000	0.2000
2	0.1800	0.1800	0.1000	0.1800	0.0000	0.1800	0.0000	0.0000	0.1800
3	0.1800	0.1800	0.0750	0.1800	0.0000	0.1800	0.0250	0.0000	0.1800
4	0.1800	0.1800	0.0500	0.1800	0.0000	0.1800	0.0500	0.0000	0.1800
5	0.1800	0.1800	0.0250	0.1800	0.0000	0.1800	0.0750	0.0000	0.1800
6	0.1800	0.1800	0.0000	0.1800	0.0000	0.1800	0.1000	0.0000	0.1800
7	0.1600	0.1600	0.2000	0.1600	0.0000	0.1600	0.0000	0.0000	0.1600
8	0.1600	0.1600	0.1500	0.1600	0.0000	0.1600	0.0500	0.0000	0.1600
9	0.1600	0.1600	0.1000	0.1600	0.0000	0.1600	0.1000	0.0000	0.1600
10	0.1600	0.1600	0.0500	0.1600	0.0000	0.1600	0.1500	0.0000	0.1600
11	0.1600	0.1600	0.0000	0.1600	0.0000	0.1600	0.2000	0.0000	0.1600
12	0.1400	0.1400	0.3000	0.1400	0.0000	0.1400	0.0000	0.0000	0.1400
13	0.1400	0.1400	0.2250	0.1400	0.0000	0.1400	0.0750	0.0000	0.1400
14	0.1400	0.1400	0.1500	0.1400	0.0000	0.1400	0.1500	0.0000	0.1400
15	0.1400	0.1400	0.0750	0.1400	0.0000	0.1400	0.2250	0.0000	0.1400
16	0.1400	0.1400	0.0000	0.1400	0.0000	0.1400	0.3000	0.0000	0.1400
17	0.1200	0.1200	0.4000	0.1200	0.0000	0.1200	0.0000	0.0000	0.1200
18	0.1200	0.1200	0.3000	0.1200	0.0000	0.1200	0.1000	0.0000	0.1200
19	0.1200	0.1200	0.2000	0.1200	0.0000	0.1200	0.2000	0.0000	0.1200
20	0.1200	0.1200	0.1000	0.1200	0.0000	0.1200	0.3000	0.0000	0.1200
21	0.1200	0.1200	0.0000	0.1200	0.0000	0.1200	0.4000	0.0000	0.1200
22	0.1000	0.1000	0.5000	0.1000	0.0000	0.1000	0.0000	0.0000	0.1000
23	0.1000	0.1000	0.3750	0.1000	0.0000	0.1000	0.1250	0.0000	0.1000
24	0.1000	0.1000	0.2500	0.1000	0.0000	0.1000	0.2500	0.0000	0.1000
25	0.1000	0.1000	0.1250	0.1000	0.0000	0.1000	0.3750	0.0000	0.1000
26	0.1000	0.1000	0.0000	0.1000	0.0000	0.1000	0.5000	0.0000	0.1000

Table 5.3: Parameter Settings (PS) from point packing with minimum distance of 0.35 to be used for Exp. **C**.

PS	Togg.	Hop	Add	Del.	Swap	L-Togg.	L-Add	L-Del.	Null
1	0.0013	0.0145	0.0374	0.0133	0.4139	0.0281	0.0107	0.2846	0.1962
2	0.5210	0.0096	0.0283	0.3316	0.0237	0.0135	0.0056	0.0218	0.0449
3	0.2528	0.0142	0.0087	0.2138	0.0021	0.2267	0.2228	0.0079	0.0509
4	0.0042	0.5606	0.0094	0.0055	0.0598	0.0113	0.0292	0.0308	0.2892
5	0.3876	0.1348	0.0250	0.0104	0.0315	0.3750	0.0085	0.0056	0.0216
6	0.0082	0.0335	0.0254	0.1409	0.2295	0.0279	0.2669	0.2420	0.0257
7	0.0317	0.0007	0.2116	0.0121	0.2291	0.1894	0.0018	0.0048	0.3187
8	0.2170	0.1965	0.0209	0.2728	0.0657	0.0031	0.0019	0.2111	0.0111
9	0.0024	0.0036	0.0074	0.0009	0.4092	0.0604	0.4846	0.0248	0.0067
10	0.0339	0.0322	0.0159	0.0004	0.0409	0.0076	0.2603	0.0036	0.6052
11	0.3136	0.0257	0.2158	0.0245	0.0005	0.0016	0.0086	0.2082	0.2015
12	0.0192	0.0322	0.4278	0.0307	0.0187	0.0531	0.1712	0.0262	0.2210
13	0.0095	0.0195	0.1936	0.2307	0.0364	0.0445	0.0129	0.2488	0.2041
14	0.0052	0.0221	0.0088	0.5403	0.3377	0.0586	0.0021	0.0058	0.0194
15	0.0026	0.4736	0.0142	0.0152	0.3205	0.1318	0.0097	0.0166	0.0158
16	0.0183	0.0171	0.1063	0.0335	0.2484	0.0033	0.2818	0.0195	0.2719
17	0.1827	0.2299	0.0068	0.0141	0.2395	0.1041	0.0219	0.0036	0.1974
18	0.1624	0.2728	0.2574	0.0020	0.0054	0.0113	0.0139	0.0169	0.2580
19	0.0703	0.0521	0.4787	0.0139	0.3289	0.0017	0.0152	0.0122	0.0272
20	0.2470	0.0190	0.2148	0.1066	0.1674	0.1864	0.0047	0.0193	0.0347
21	0.0383	0.4753	0.0165	0.0204	0.0139	0.0106	0.3416	0.0345	0.0488
22	0.0181	0.0288	0.0008	0.0302	0.4775	0.3370	0.0125	0.0380	0.0571
23	0.0100	0.0289	0.0168	0.3018	0.0063	0.2717	0.0456	0.0284	0.2906
24	0.3227	0.4718	0.0221	0.0598	0.0086	0.0053	0.0010	0.0227	0.0861
25	0.0630	0.0131	0.0320	0.0003	0.0186	0.0022	0.0079	0.2790	0.5839
26	0.0056	0.0016	0.0133	0.0032	0.0272	0.0177	0.8115	0.0214	0.0985
27	0.0097	0.0311	0.0641	0.0184	0.0068	0.2727	0.2621	0.0303	0.3048
28	0.4752	0.0075	0.0069	0.0940	0.0019	0.0125	0.0031	0.3606	0.0382
29	0.3052	0.0039	0.0032	0.0068	0.0029	0.0142	0.0051	0.0204	0.6381
30	0.7537	0.0097	0.0551	0.0129	0.0286	0.0310	0.0255	0.0437	0.0398
31	0.0299	0.0521	0.2231	0.0091	0.2253	0.0000	0.0063	0.4334	0.0209
32	0.1499	0.0108	0.2576	0.2280	0.0384	0.0043	0.2632	0.0144	0.0333
33	0.2713	0.0041	0.0133	0.0129	0.0311	0.0446	0.5675	0.0068	0.0484
34	0.0052	0.5016	0.0216	0.0649	0.0003	0.3428	0.0086	0.0029	0.0520
35	0.0122	0.0562	0.0077	0.8188	0.0215	0.0182	0.0159	0.0105	0.0390
36	0.0200	0.0163	0.3498	0.0067	0.0303	0.0047	0.0019	0.0127	0.5575
37	0.0014	0.0037	0.0153	0.2139	0.0052	0.5059	0.1873	0.0102	0.0571
38	0.2625	0.0753	0.0296	0.0183	0.2670	0.0223	0.2740	0.0269	0.0241
39	0.0474	0.0448	0.0220	0.0186	0.8183	0.0039	0.0208	0.0053	0.0189
40	0.0171	0.0612	0.0321	0.0082	0.0000	0.4944	0.0092	0.3120	0.0657

Table 5.4: Parameter Settings (PS) from point packing with minimum distance of 0.35 to be used for Exp. **C**.

PS	Togg.	Hop	Add	Del.	Swap	L-Togg.	L-Add	L-Del.	Null
41	0.2067	0.0029	0.0122	0.2816	0.2316	0.0215	0.0919	0.0122	0.1396
42	0.0178	0.0342	0.0395	0.0172	0.1119	0.5079	0.0029	0.0152	0.2532
43	0.2574	0.1292	0.0079	0.0056	0.0002	0.0042	0.1834	0.0700	0.3422
44	0.0110	0.0630	0.4820	0.0419	0.0261	0.3131	0.0071	0.0381	0.0178
45	0.0267	0.2889	0.0032	0.5082	0.0014	0.0146	0.0687	0.0600	0.0284
46	0.0301	0.0301	0.0071	0.4044	0.0573	0.0039	0.4091	0.0301	0.0278
47	0.0569	0.3108	0.4790	0.0358	0.0080	0.0128	0.0154	0.0723	0.0091
48	0.0333	0.2106	0.0077	0.0519	0.1585	0.3089	0.1942	0.0064	0.0284
49	0.0044	0.0419	0.0082	0.0149	0.0135	0.3141	0.5054	0.0366	0.0611
50	0.0009	0.2389	0.0057	0.0289	0.0257	0.0015	0.2376	0.2397	0.2211
51	0.0821	0.0058	0.4506	0.0577	0.0032	0.0217	0.0515	0.2980	0.0292
52	0.2732	0.0203	0.0281	0.0283	0.0088	0.2954	0.0009	0.0059	0.3392
53	0.0111	0.0214	0.0174	0.0152	0.0125	0.0001	0.0086	0.8152	0.0985
54	0.0002	0.0537	0.0158	0.0841	0.0350	0.2202	0.0004	0.5641	0.0266
55	0.5538	0.0226	0.0153	0.0031	0.3259	0.0139	0.0196	0.0206	0.0251
56	0.0023	0.0460	0.0106	0.0162	0.0154	0.2868	0.0075	0.2704	0.3448
57	0.0104	0.2906	0.2388	0.0064	0.2286	0.0055	0.1638	0.0217	0.0342
58	0.2768	0.0041	0.0092	0.0202	0.0046	0.1133	0.2531	0.2810	0.0376
59	0.0972	0.0064	0.0078	0.4710	0.0035	0.3930	0.0006	0.0134	0.0071
60	0.0358	0.2520	0.1084	0.2817	0.0025	0.2724	0.0053	0.0122	0.0297
61	0.0049	0.0111	0.2527	0.0003	0.3048	0.2133	0.1832	0.0089	0.0208
62	0.0026	0.0376	0.0401	0.2856	0.2110	0.2173	0.0020	0.1825	0.0213
63	0.0046	0.0398	0.1853	0.1300	0.5136	0.0400	0.0105	0.0448	0.0315
64	0.0033	0.0145	0.0048	0.0262	0.0018	0.0256	0.5288	0.0175	0.3774
65	0.0309	0.0066	0.0121	0.0100	0.0148	0.8240	0.0210	0.0067	0.0741
66	0.1586	0.0047	0.1787	0.5368	0.0212	0.0412	0.0020	0.0258	0.0310
67	0.0056	0.0219	0.0177	0.4372	0.0255	0.0147	0.0105	0.4473	0.0196
68	0.0266	0.0188	0.0029	0.0033	0.0665	0.0211	0.0054	0.0002	0.8552
69	0.0156	0.4267	0.0049	0.0292	0.0319	0.0105	0.0206	0.4256	0.0348
70	0.0072	0.5227	0.0999	0.2535	0.0055	0.0030	0.0690	0.0017	0.0373
71	0.0089	0.0002	0.3233	0.0183	0.0020	0.0128	0.4968	0.0249	0.1128
72	0.0141	0.0061	0.0105	0.0114	0.3714	0.0008	0.0136	0.0481	0.5239
73	0.0075	0.2373	0.2009	0.0070	0.0483	0.2103	0.0316	0.2377	0.0194
74	0.3296	0.0009	0.4767	0.0360	0.0263	0.0369	0.0111	0.0012	0.0811
75	0.4925	0.0061	0.0120	0.0517	0.0083	0.0052	0.2846	0.0127	0.1268
76	0.0424	0.0519	0.0112	0.5233	0.0174	0.0070	0.0099	0.0096	0.3273
77	0.2812	0.0270	0.0030	0.0538	0.2586	0.0951	0.0031	0.2592	0.0191
78	0.0046	0.1773	0.0649	0.0990	0.1194	0.0323	0.4604	0.0056	0.0365
79	0.0481	0.3252	0.0009	0.0203	0.0009	0.0301	0.0230	0.0085	0.5430
80	0.0071	0.0223	0.0159	0.0048	0.0002	0.3492	0.0003	0.0201	0.5801

Table 5.5: Parameter Settings (PS) from point packing with minimum distance of 0.35 to be used for Exp. **C**.

PS	Togg.	Hop	Add	Del.	Swap	L-Togg.	L-Add	L-Del.	Null
81	0.0197	0.0795	0.7238	0.0001	0.0481	0.0175	0.0393	0.0038	0.0684
82	0.0164	0.7935	0.0291	0.0074	0.0499	0.0299	0.0308	0.0329	0.0100
83	0.0084	0.0159	0.0323	0.0172	0.0021	0.0046	0.4702	0.3775	0.0718
84	0.0586	0.0304	0.4598	0.3598	0.0204	0.0155	0.0272	0.0107	0.0175
85	0.0200	0.0130	0.0721	0.2606	0.0071	0.0326	0.0052	0.0484	0.5409
86	0.0876	0.0420	0.0108	0.0061	0.0690	0.0213	0.0032	0.4764	0.2836
87	0.2740	0.0123	0.0136	0.0024	0.5226	0.0142	0.0491	0.0042	0.1075
88	0.0017	0.2278	0.1060	0.2606	0.0284	0.0043	0.1286	0.0060	0.2365
89	0.5255	0.0053	0.0375	0.0014	0.0560	0.0315	0.0056	0.0120	0.3252
90	0.4981	0.2045	0.2272	0.0210	0.0135	0.0170	0.0013	0.0026	0.0147

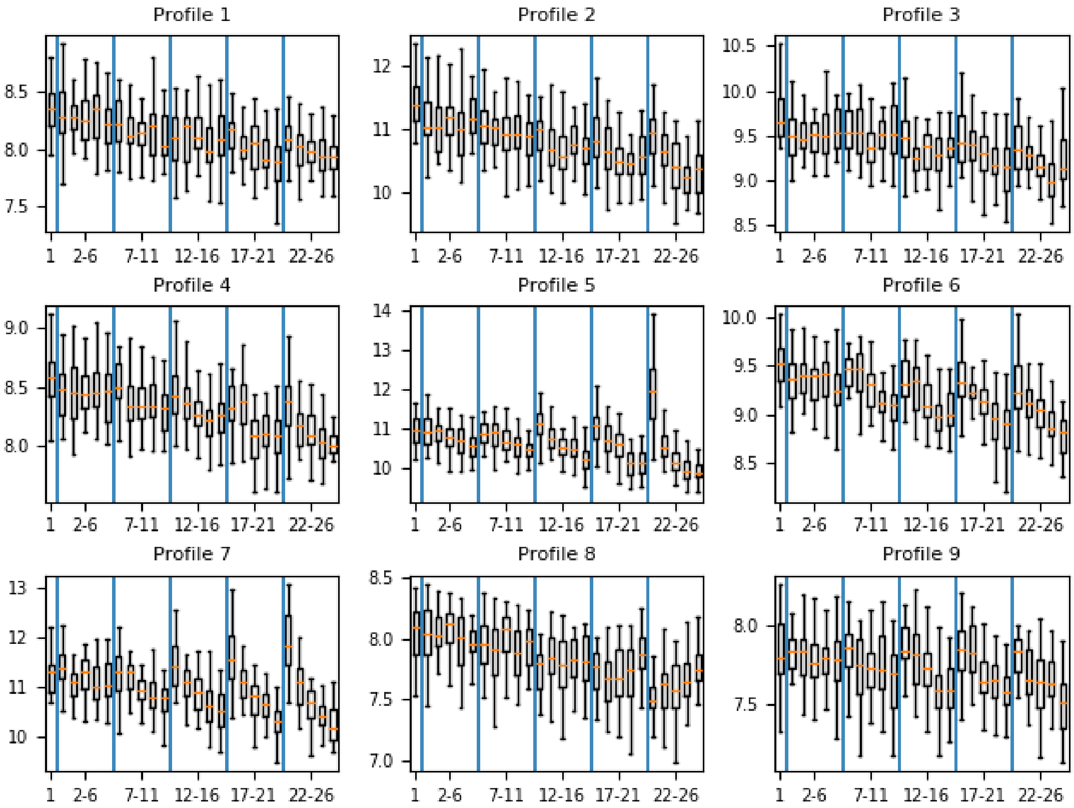
to generate the initial population of PSs and as the variation operator used in the evolutionary algorithm; see further details in [5]. To run this algorithm, a researcher must choose a value for the minimum allowable distance between PSs. The minimum distance used here is 0.35 which returned 81-90 PSs, as found in Table 5.3. This minimum distance was chosen at it allowed testing of a reasonably larger number of PSs than had been previously used for point packing.

5.3.3 Exp. D: The SIRS Model on the ED Problem

The ED problem was reintroduced to this body of research to determine the impact on the representation from the new operations added since [45]. This addition also allows for a natural introduction of the SIRS epidemic model as the change will ultimately increase the length of epidemics. Now an epidemic has the potential to last indefinitely, although this did not happen for any epidemics simulated within this paper. To explore both of these additions a small point packing, using the process from Section 5.3.2, was conducted with a minimum distance of 0.535 (see Table 5.6). This distance was chosen as it produced a similar number of PSs as in Section 5.3.1. To explore the impact of the SIRS model compared to the SIR model, the number of time steps during which an individual remains removed was set to 6, 8, and 10. Therefore, the four different environments used to test the ED problem are SIR, SIRS 6, SIRS 8, and SIRS 10.

Table 5.6: Parameter Settings (PS) from point packing with minimum distance of 0.535 to explore SIR(S) on Exp. **D**.

PS	Togg.	Hop	Add	Del.	Swap	L-Togg.	L-Add	L-Del.	Null
1	0.0312	0.3528	0.2545	0.2835	0.0065	0.0151	0.0011	0.0324	0.0229
2	0.3988	0.0250	0.4167	0.0122	0.0380	0.0016	0.0325	0.0569	0.0182
3	0.0107	0.0358	0.7801	0.0243	0.0614	0.0563	0.0130	0.0057	0.0128
4	0.0438	0.0188	0.0281	0.0499	0.0003	0.4322	0.0051	0.0170	0.4047
5	0.0085	0.0104	0.0232	0.0177	0.4035	0.0008	0.0613	0.4326	0.0421
6	0.7592	0.0305	0.0024	0.0060	0.0068	0.0344	0.0106	0.1176	0.0326
7	0.0122	0.0196	0.3838	0.0942	0.0070	0.0067	0.0303	0.0173	0.4288
8	0.0020	0.0727	0.0010	0.0435	0.0355	0.0123	0.7946	0.0201	0.0184
9	0.0073	0.0261	0.3406	0.0015	0.0195	0.3741	0.0077	0.2037	0.0195
10	0.4303	0.0847	0.0105	0.0086	0.4158	0.0042	0.0025	0.0208	0.0226
11	0.0108	0.0072	0.0030	0.0617	0.4359	0.0010	0.0129	0.0223	0.4451
12	0.3682	0.0191	0.0319	0.3670	0.0191	0.1107	0.0127	0.0466	0.0247
13	0.0199	0.0032	0.3243	0.1881	0.3991	0.0012	0.0262	0.0269	0.0110
14	0.0180	0.0491	0.0002	0.7646	0.0277	0.0174	0.0661	0.0047	0.0522
15	0.0334	0.4149	0.0113	0.0005	0.0323	0.4130	0.0213	0.0373	0.0360
16	0.0129	0.3896	0.0193	0.0034	0.0038	0.0001	0.1483	0.4096	0.0130
17	0.0093	0.0211	0.0098	0.0020	0.8084	0.0532	0.0000	0.0519	0.0442
18	0.0341	0.0192	0.0044	0.2095	0.0586	0.3328	0.3233	0.0047	0.0134
19	0.0531	0.7894	0.0394	0.0261	0.0114	0.0203	0.0150	0.0164	0.0289
20	0.0038	0.0457	0.4066	0.0768	0.0226	0.0061	0.3952	0.0268	0.0164
21	0.0032	0.3982	0.0129	0.0426	0.0336	0.0115	0.0070	0.0539	0.4371
22	0.0259	0.0140	0.0032	0.2220	0.0056	0.0160	0.0203	0.3623	0.3308
23	0.0053	0.0229	0.0372	0.0192	0.0110	0.0134	0.0435	0.0045	0.8430
24	0.0176	0.3765	0.0125	0.0156	0.3637	0.0094	0.1757	0.0099	0.0190
25	0.3572	0.2016	0.0045	0.0079	0.0026	0.0088	0.3556	0.0510	0.0109
26	0.0027	0.0091	0.0011	0.0628	0.0071	0.8545	0.0484	0.0028	0.0115
27	0.0038	0.0156	0.0221	0.0021	0.0285	0.0267	0.4183	0.0396	0.4432
28	0.0265	0.0456	0.0144	0.0109	0.0179	0.0102	0.0087	0.7966	0.0691
29	0.4238	0.0343	0.0021	0.0319	0.0021	0.0066	0.0487	0.0037	0.4469



[t]

Figure 5.1: Box and whisker plots of the 30 PM-Fitness results from Exp. **A**. Vertical lines added to aid readability and coincide with horizontal lines in Table 5.2. Lower fitness is better.

5.4 Results and Discussion

A brief summary of the results from the four experiments are provided in Table 5.7. Each of these are discussed in more detail in the following sections.

5.4.1 Exp. A: Local Add on PM Problem

Box and whisker plots of the fitness values achieved in each of the 26 PSs for testing local add against the original implementation are displayed within Fig. 5.1. For

Table 5.7: A brief summary of results from the four experiments.

Exp.	**Testing**	**Summary of Findings**
A	Local Add	Better fitness on all but profile 8
B	Local Delete	Worse fitness on all profiles
C	Point Packing	Best fitness values on all profiles
D	SIRS Model	Longer, more volatile epidemics

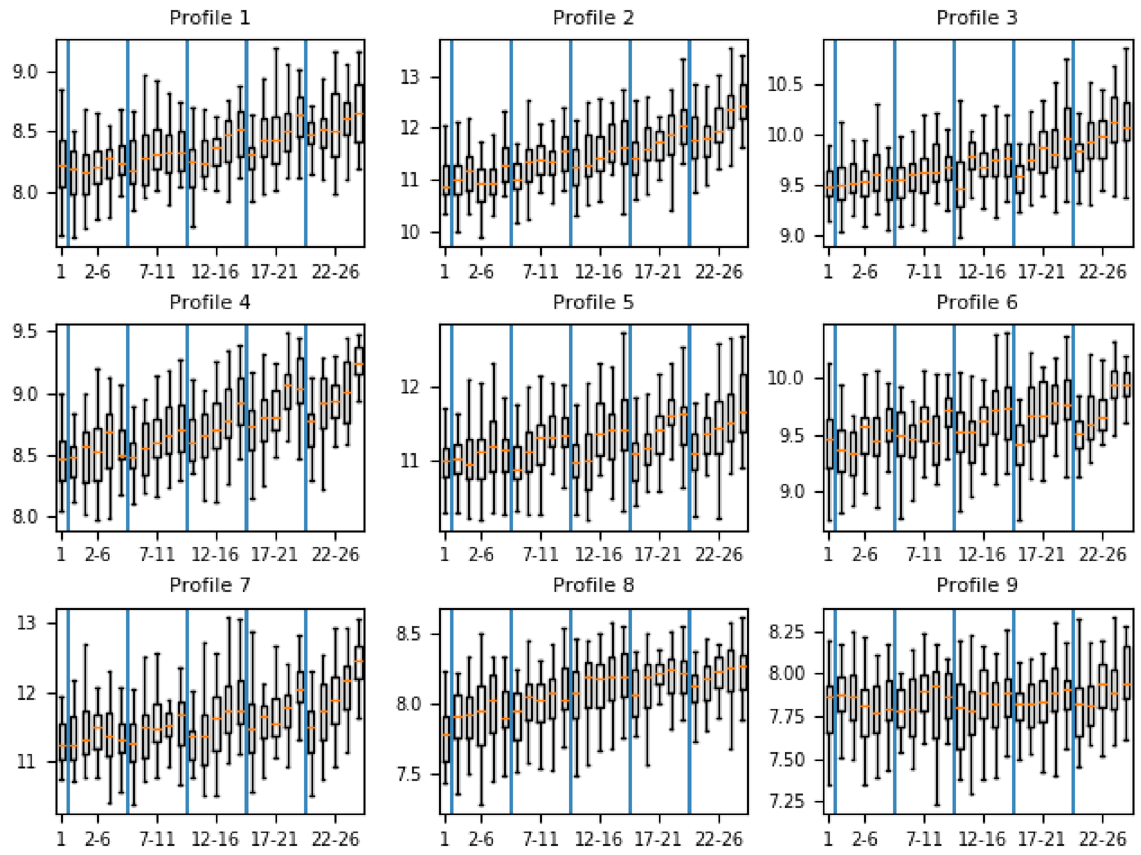


Figure 5.2: Box and whisker plots of the 30 PM-Fitness results from Exp. **B**. Vertical lines added to aid readability. Lower fitness is better.

most profiles there is an obvious skew towards a lower, or better, fitness value as the overall use of add increases regardless of the variant of add in question. Furthermore, most profiles benefit from the replacement of add with local add. This phenomena is most obvious in profiles 5, 6, and 7. In contrast, profile 8 does not follow the trend observed in the other profiles. This fact is most evident in PS 22 to 26 where there is a upward trend as the use of add is replaced with local add. A caveat to these findings is that these are slight trends in that the confidence intervals of all PSs within a given profile overlap. These findings correlate with those found in [20] where local toggle performed better on all profiles except profile 8. This provides further evidence that better performance may be achieved if parameters are tuned on a per-profile basis.

5.4.2 Exp. B: Local Delete on PM Problem

Box and whisker plots of the fitness values achieved in each of the 26 PSs for testing local delete against the original implementation are displayed within Fig. 5.2. All the profiles show a decline in performance as the overall use of delete increases regardless of variant. Additionally, the replacement of delete with its local variant results in worse fitness for most profiles. This trend is seen most apparently in profiles 4, 5, 6 and 7. With local delete, all profiles receive decreased fitness as delete is replaced with its local variant. The confidence intervals for each of the PSs on a given profile mostly overlap in a similar fashion to the local add results demonstrating that the trends observed are minor.

5.4.3 Exp. C: Point Packing on PM Problem

The results from the different profiles used provides adequate evidence to suggest that some profiles are impacted significantly by the choice of parameters while others exhibit better consistency regardless of PS. For example, looking at the results for profiles 7 and 8 in Fig. 5.3 the same PSs resulted in a fitness range of 8.96-39.68 in profile 7 while in profile 8 the range was 6.57-9.14. These results point to the previous observation that performance of the evolutionary algorithm is significantly impacted by the profile being used to perform the evolutionary algorithm. Additionally, some PSs have desirable results on some profiles while at the same time have horrible results on other profiles. This is most notably observed in PSs 30, 74, 81, and 90. It is also worth noting that the point packing parameter exploration returned PSs which resulted in fitness values far better than all those observed in both traditional parameter sweeps for all the profiles. Additionally, these fitness values obtained are of statistical significance since for all of the profiles, there is no overlap in the confidence intervals.

To provide additional insight Fig. 5.4 provides a plot of the best, worst and median rank achieved by each of the 90 point packing PSs, sorted by median. This figure clearly demonstrates that there are PSs that perform well on all epidemic profiles used within this study. The three best PSs, namely 26, 33 and 71, all have large probabilities associated with the local version of the add operator. The proportions assigned to local add for those three PSs are 81%, 57%, and 50% respectively. This provides further evidence to suggest that the local add operator is superior to the original implementation.

5.4.4 Exp. D: SIRS Model on ED Problem

The ED-Fitness values, with 95% confidence interval, for the 29 PSs can be found in Fig. 5.5. PSs 20, 23, and 27 yield good results across the four environments

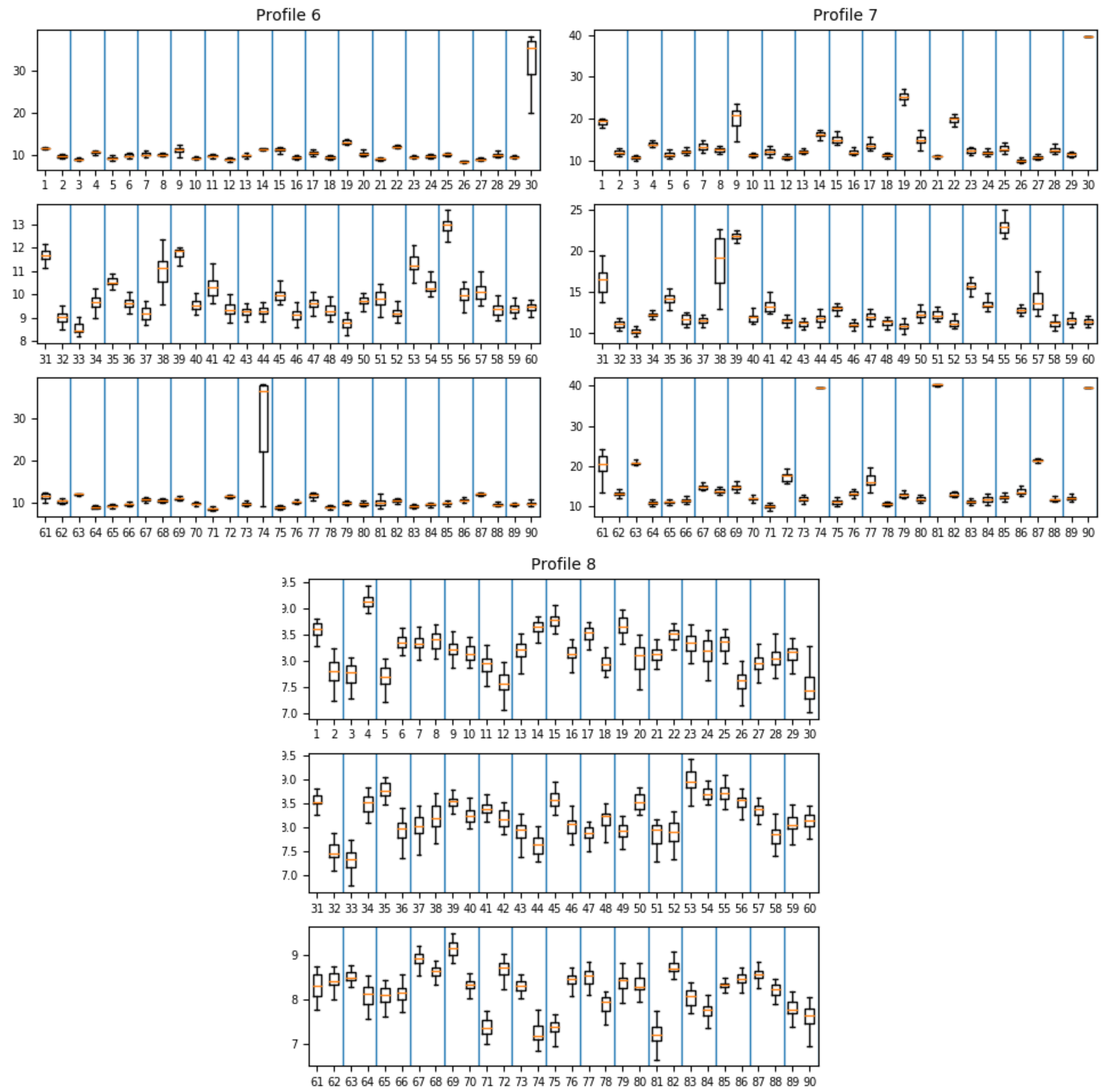


Figure 5.3: Box and whisker plots of the 30 PM-Fitness results for the 90 paramater setting in Exp. **C**. Vertical lines added to aid readability and coincide with horizontal lines in Table 5.6. Lower fitness is better.

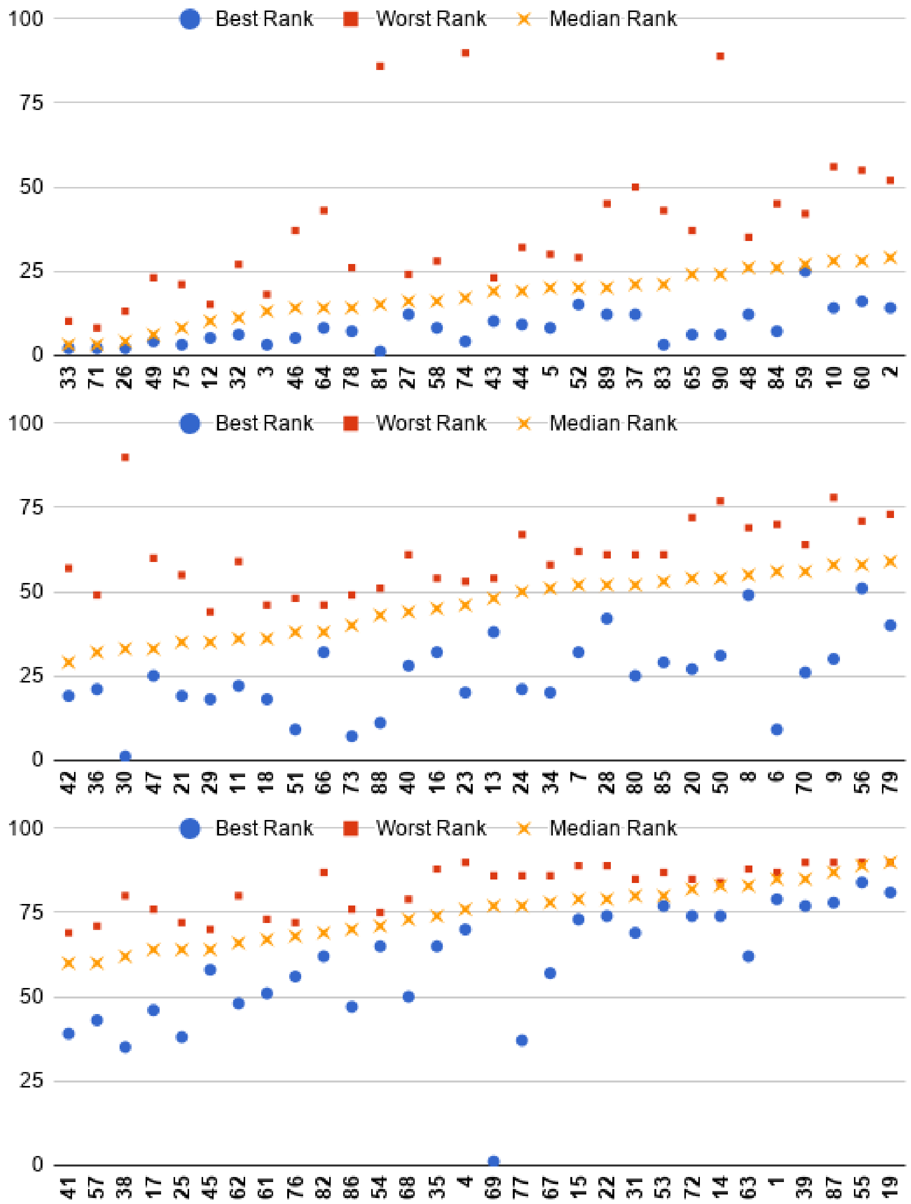
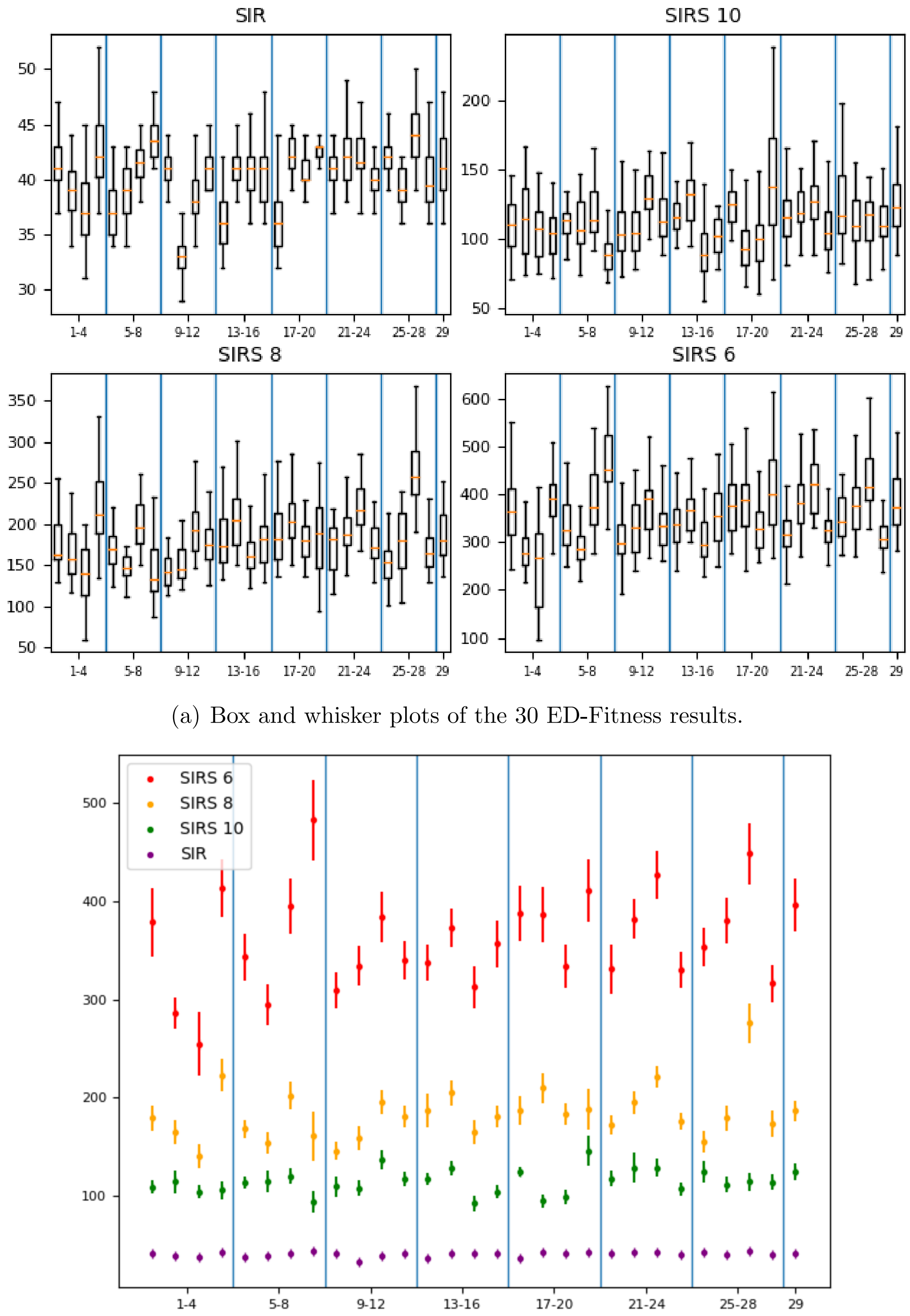


Figure 5.4: Best, median and worst rank achieved from all nine profiles in Exp. **C** for each of the point packing PSs using PM-fitness, sorted by increasing median. Lower rank is better.



(a) Box and whisker plots of the 30 ED-Fitness results.

(b) The mean ED-Fitness with 95% confidence interval.

Figure 5.5: Results from Exp. **D** within the four environments: SIR, SIR 6, SIR 8, and SIR 10. Vertical lines added to aid readability and coincide with the horizontal lines in Table 5.6. Higher fitness is better.

used for testing. PS 20 consisted of mainly add and local add; it is common for a point packing to include an PS exhibiting these characteristics to be among the best PSs, regardless of problem, see [20]. Both PSs 23 and 27 have large amounts of null probability with PS 27 also having a significant portion attributed to local add. This could indicate that a graph closer to the starting graph provides better fitness for the ED problem. PSs 2, 3, and 6 all performed poorly across the environments. These PSs were dominated by the toggle and/or add operations, each providing more than 75% of the proportion to toggle and add combined. An interesting observation for the SIRS 6 series in Fig. 5.5(b) is that the standard deviation is closely correlated with the overall performance of a PS. This is true for all PSs in the SIRS 6 environment other than PS 3, which is dominated by the add operation. Also, PS 8 performs remarkably with the highest mean epidemic duration of 483 time steps and the best fitness of 939 time steps, a value 235 above any network found by any other PS. In contrast, PS 8 has sub-optimal results for SIRS 8 and SIRS 10. PS 8 also consistently has a large standard deviation compared to the other PSs in all environments.

Regardless of PS there are clear trends moving from the SIR model towards the SIRS 10, 8, and 6 models. Firstly, the standard deviation elevates dramatically, ranging between 1.40-3.55 in the SIR model to 44.3-115.5 with SIRS 6. This means the ability for a PS to result in stable network behaviour between multiple epidemic simulations is impacted greatly. Secondly, as expected the ability for individuals to be susceptible a second time has dramatically increased the length of epidemics observed. The average epidemic length progressed from 40.4, to 114.6, 183.2, and finally 361.2. This value will increase to the point of epidemics which last infinitely long due to the definition used for epidemic length in this paper and the permission of cycles within the graph.

5.5 Conclusions and Future Work

The findings of this paper as well as [3], [45] and [20] indicate that three of the four new local operators resulted in better fitness for all nine epidemic profiles. Therefore, moving forward there is strong evidence to suggest that the hop, local toggle, and local add operations belong within the generative representation discussed in this paper. The local delete operation can be kept for completeness though should be set to 0% when performing traditional parameter sweeps.

Chapter 6

Case Study 2

Pandemic: A Graph Evolution Story

This case study was originally published in CIBCB 2019 conference proceedings [24]. The Graph Evolution Tool (GET) was built to generate personal contact networks representing who can infect whom within a community. The tool is expanded in order to permit an infection scheme which divides the community into different districts, thus permitting within-district and between-district infections. The evolutionary algorithm comprising GET is expanded upon to simulate communities which include 512 individuals in up to eight districts, initially infecting one person in one district and spreading through a community. The overall goal is to generate communities that will maximize the length of an epidemic. The problem associated with adequately exploring the numerous parameters accompanying evolutionary algorithms is addressed using a point packing and insight from previous work. The Susceptible-Infected-Removed (SIR) model of infection was chosen as it provides a sufficient balance of simplicity and complexity for the problem.

6.1 Introduction

In [26], the researchers use personal contact networks to represent a human population and compare simulated epidemics against data from an outbreak of H1N1 in 2009 on a university campus. Their personal contact network is generated in a series of layers with each layer adding more edges to the network. The first layer has edges between members living in the same residence room. The second layer connects members who are enrolled in the same class. Successive layers represent less intense and less frequent interactions between individuals (e.g. attending the same university). The network begins as vertices without any edges and the layers are applied such that the graph is connected only after the final layer is applied. Thus, the final layer adds edges between several sub-networks which share no edges between one-another.

The current study combines a similar network structure as [26] with the Local THADS-N generative representation. The problem of maximizing epidemic duration is used to test personal contact networks that are arranged into a set of sub-networks. This arrangement is analogous to a set of districts or neighbourhoods, each with separate personal contact networks.

Within a district individuals have frequent interactions, while there are less frequent interactions with nearby districts such as when an individual travels from their own neighbourhood to another. The study thus considers how the length of the epidemic is affected when an epidemic starts in one district and then with some probability ”jumps” to another district.

The remainder of this paper is organized as follows. Section 4.1 formally defines the representation. Section 5.3 reviews the design of the experiments, how parameters are obtained and their values. The results from the experiments will be presented and analyzed in Section 5.4. Lastly, Section 5.5 will conclude the paper as well as present possibilities for future research.

6.2 The Local THADS-N Representation

Recall that the Local THADS-N representation does not directly specify a solution to a problem. Instead, the representation provides several *editing operations* that can be used to modify a graph. A solution to the problem is then specified by applying a sequence of edits to the initial graph.

In this study, the strings of operations are applied to an initial graph that has the general structure shown in Fig. 4.1. The example shown has 128 vertices but it is

Table 6.1: The sets of Local THADS-N operation densities used for constructing solutions to the epidemic duration problem

Set	Toggle	Hop	Add	Delete	Swap	L-Toggle	L-Add	L-Delete	Null	Justification for Choice
1	0.04381	0.01885	0.02812	0.04993	0.00027	0.43224	0.00508	0.01696	0.40473	High Fitness
2	0.00382	0.01564	0.02205	0.00214	0.02848	0.02672	0.41832	0.03961	0.44321	High Fitness
3	0.03415	0.01919	0.00443	0.20950	0.05859	0.33277	0.32325	0.00469	0.01343	High Fitness
4	0.01763	0.37651	0.01251	0.01565	0.36368	0.00940	0.17572	0.00993	0.01898	Low Standard Deviation
5	0.35718	0.20156	0.00448	0.00787	0.00263	0.00880	0.35559	0.05101	0.01090	Low Standard Deviation
6	0.01221	0.01963	0.38384	0.09418	0.00695	0.00673	0.03030	0.01730	0.42884	Low Standard Deviation
7	0.00377	0.04566	0.40664	0.07685	0.02258	0.00606	0.39520	0.02680	0.01644	Both
8	0.00197	0.07271	0.00097	0.04354	0.03551	0.01226	0.79455	0.02013	0.01836	Both

generalized to graphs with the desired number of vertices for a given district. This structure was chosen as previous research [8, 11] demonstrated that graphs having vertices with four or five edges are desirable for the test problem at hand.

One possible representation for directly evolving graphs with n vertices would be a binary gene with $\binom{n}{2}$ loci, each of which specifies the presence or absence of a possible edge. This representation seems natural. However, it was found to perform badly on many test problems [10], as were its subsequent modifications. The reason for this poor performance relates to the fact that almost all interesting graphs are sparse, while very few graphs represented in this manner are sparse. Although it is possible to adjust the primitive probability of an edge existing, this helps only in a limited way because in this representation crossover has a high probability of increasing the number of edges when applied to two sparse graphs.

The generative representation used in this study also searches a relatively restricted portion of network space, namely a sphere, in the space, whose centre is the starting graph and with radius equal to the number of allowed edits.

6.2.1 Past solutions to the epidemic duration problem

Recall that networks yielding a long epidemic duration were found to be ”banana” shaped. An example of this shaped network is included in Figure 6.4(a). In this work we would expect individual districts to become banana shaped.

6.3 Experimental Design

6.3.1 Point Packing Parameter Selection

The current study relies on sets of parameters from an earlier study [21]. These parameters were determined by point packing from Section 4.5, a mechanism which

Table 6.2: The variables associated with the various epidemic communities for the epidemic duration problem

Community	k	district Size	Total Vertices	α'
A	1	512	512	N/A
B	4	128	512	0.10
C	4	128	512	0.05
D	8	64	512	0.10
E	8	64	512	0.05
Original	1	128	128	N/A

thoroughly explores the parameter space without the overhead of a full factorial exploration. The parameter sets used here were selected as they were the top-performing from amongst those generated, which also considered the problem of maximizing epidemic duration, albeit for a different epidemic model and only for single communities of a fixed size. A minimum distance of 0.535 was specified to run the point packing, generating some 29 parameter sets.

6.3.2 Description of Experiments

Eight sets of parameters are used in the current study. Of the 29 parameter sets from the previous study, we selected the five that produced results with the highest fitness, and the five that produced results with the lowest standard deviation resulting from 30 runs in the previous study. Listed in Table 6.1, it can be seen that two parameter sets satisfied both of these requirements, therefore there are eight parameter sets in total.

The densities for a given parameter set are designed to add up to 1.0. It may appear that there are some rather unusual choices of values for parameters in comparison to a traditional parameter sweep. For example, in set 1 toggle has a probability (density) of 4.381% of being selected for any given entry in the chromosome. A more traditional parameter sweep would be more likely to choose probabilities that were (for example) multiples of 5 or 10%. This type of situation, however, is a feature of point packing and it is one of the reasons it is able to explore the parameter space so well.

Each of the eight parameter settings is applied to five communities, listed in Table 6.2 as communities A–E. The overall number of individuals in a community is held constant at 512. Each of these individuals belongs to one of 8 separate districts each of size 64 (communities D and E), 4 separate districts each of size 128 (communities

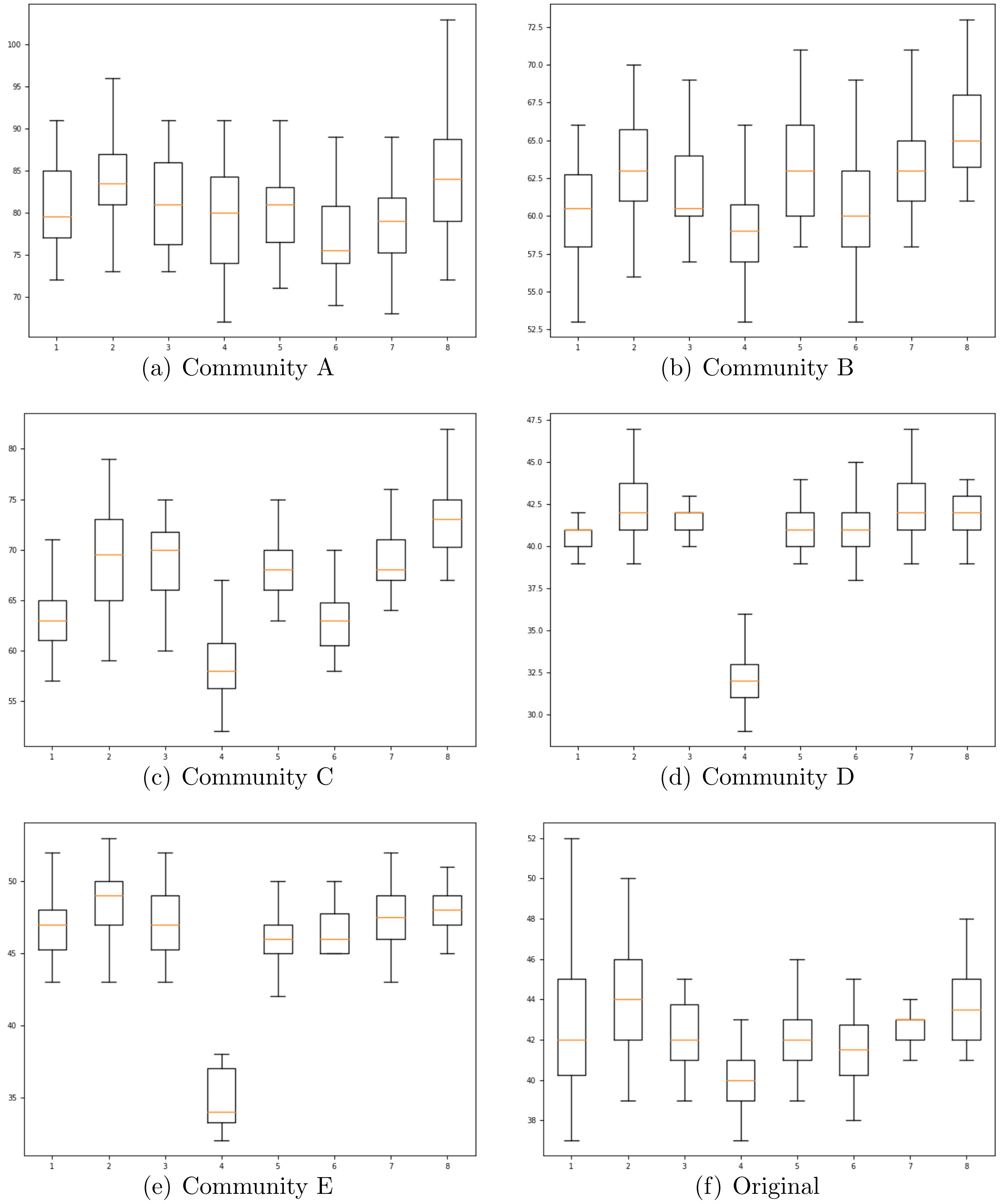
B and C), or a single district of size 512 (community A). For all communities, the probability of within-district infection is held constant at $\alpha = 50\%$. The probability of between-district infection is either $\alpha' = 5\%$ or 10% for 4 or 8 districts, and does not apply in community A as this consists of only one district. For comparison purposes, we also consider the problem of epidemic duration on a community with a single district of size 128 (listed as "original" in Table 6.2).

6.4 Results and Discussion

The results shown in Fig. 6.1 represent the box and whisker plots of the maximum epidemic lengths achieved by 30 runs of the evolutionary algorithm. Along with these, the results achieved from epidemics on a community comprised of a single district with 128 vertices is included within Fig. 6.1(f) and labelled as "original". This is provided as a comparison against the five communities totalling 512 individuals each (communities A–E) implemented in this study, allowing some insight as to what happens within a single district which is of average size in comparison to the others.

A rough sine wave pattern can be seen between parameter sets across the communities, although the variance from this pattern is observed most dramatically when four districts are used, as in communities B and C. In contrast, using eight districts flattened the observed pattern between parameter sets. One outlier, dominated by the hop and swap operations, is parameter set 4; in this case, the number of districts has a negative impact on the fitness achieved regardless of the value of α'. Parameter sets 2 and 8 each result in the best values for three of the communities; Original, D, and E yielded the best results with parameter set 2 and the rest did so with parameter set 8. As parameter set 2 relies on local add and null a large number of local connections will exist between district members when compared to those without these conditions. Parameter set 8 uses mostly local add, with marginal probability attributed to the other edge operations. Parameter settings with a sizeable use of the local add operation have been among point packing's most fit results since its introduction to the representation in [19, 21].

The epidemic data from the communities is combined into Fig. 6.2. Most glaringly, it is apparent that, of the communities tested, the optimal one for maximizing epidemic duration placed all individuals into a single district. This is based on the fact that community A clearly outperforms all the other communities tested. The addition of separate districts was expected to allow the epidemic to spread more slowly; ideally, the epidemic would spread to another district just as it had infected the last



(a) Community A

(b) Community B

(c) Community C

(d) Community D

(e) Community E

(f) Original

Figure 6.1: Box and whisker plots of the maximum epidemic duration from 30 runs of the evolutionary algorithm across all communities and all operation probability densities.

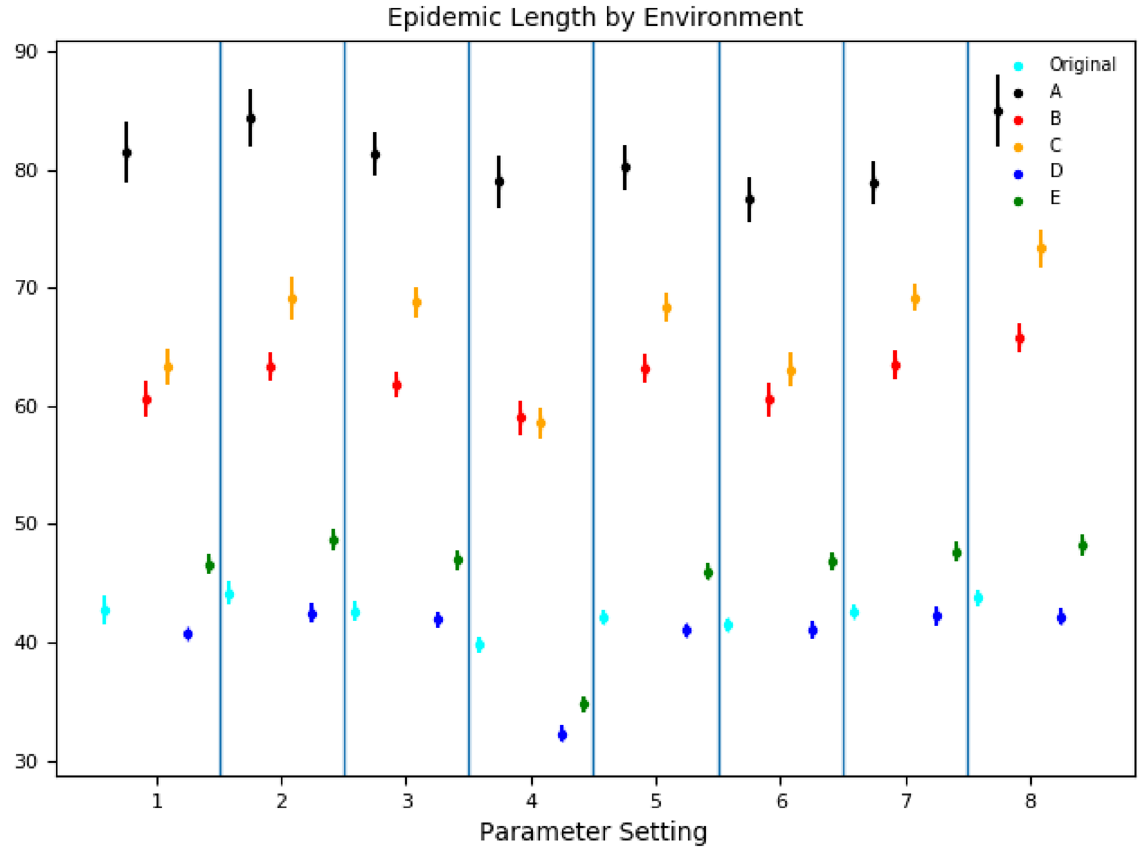


Figure 6.2: The mean of the maximum epidemic duration, with 95% confidence interval, from 30 runs of the evolutionary algorithm for all communities.

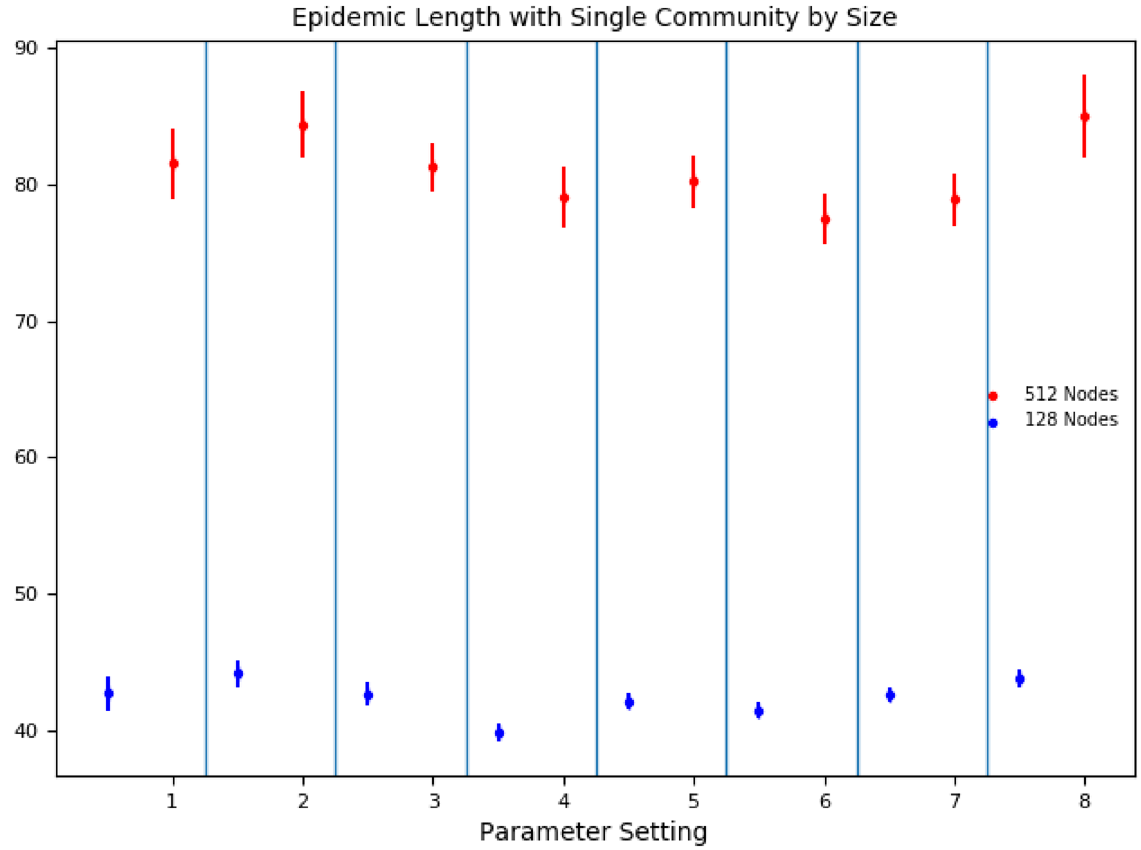
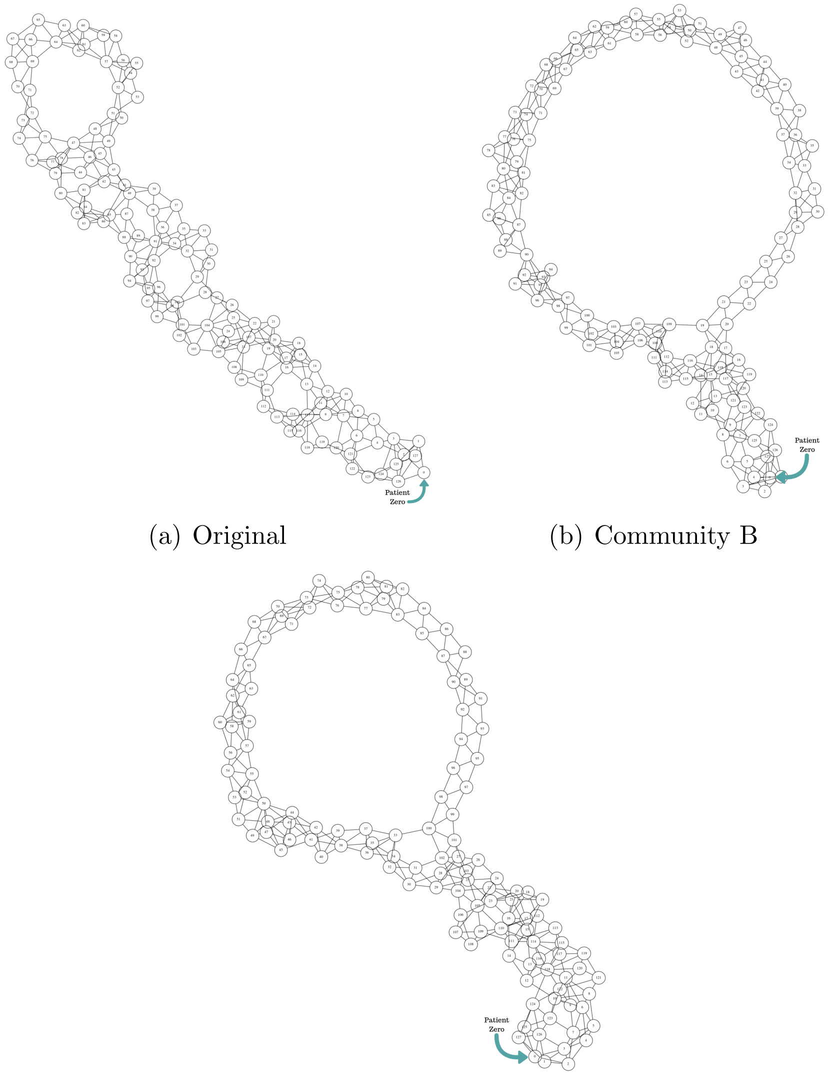


Figure 6.3: The mean of the maximum epidemic duration, with 95% confidence interval, from 30 runs of the evolutionary algorithm on communities with a single district.

members of its current district. This situation, however, did not materialize; likely this was due to a value of α' which was too high, resulting in the epidemic spreading too quickly between districts. This is evidenced by the fact that decreasing the value of α' provided significant fitness increases for all but one (parameter set, community) pair, namely parameter set 4 when comparing communities B (with $\alpha' = 0.10$) and C (with $\alpha' = 0.05$ and all other settings identical to B). The significant impact observed by changing the values of α', the number of districts, and the district size indicate that further optimization is required in order to best pair community settings with probability densities.

Furthermore, Fig. 6.3 features two communities, each comprised of a single district with either 128 or 512 vertices. This was included to demonstrate how a change to only the district size can impact the fitness evaluation. Obviously, the larger district is able to sustain a longer epidemic as there are more individuals to infect, so the increase is unsurprising. However, note that the four-fold increase in the number of vertices, from 128 to 512, only results in an approximately two-fold increase in epidemic duration, from 39.83–44.17 for 128 vertices to 77.5–85.03 for 512 vertices. It is also important to note that any increase in duration also coincides with an increase in standard deviation, as a wider range of possibilities are present regardless of other factors being considered.

Lastly, Fig. 6.4 provides visual representations of three districts which achieved the best fitness for communities with districts of size 128 using parameter setting 8. Recall that the three districts represented in the figure differ in both their α' values and fitness evaluations. The original community, with a single district, will have evolved differently than those within communities B and C as the fitness value depends on the length of an outbreak within one district as compared to four in communities B and C. The personal contact network from previous work appears to have diverged more from the initial ring structure than the districts from B and C; these networks have a significant area of their networks roughly resembling the starting ring. In contrast, the original district resembles the "banana" shape which has been a common trend within previous work, though there still exists a section of the network with a ring shape. Looking closely at the networks from the districts for communities B and C can provide some insight into how α', which is the probability of transmission between districts, impacts network evolution. Community B's district, with $\alpha' = 0.10$, features a smaller tail and a larger proportion of the population still within the original ring structure as compared to community C with $\alpha' = 0.05$. Looking again at the original community, which has no α' value as it is the entire



(a) Original

(b) Community B

(c) Community C

Figure 6.4: The personal contact networks of size 128 representing the district which resulted in the longest epidemic with parameter set 8 within the designated community. Patient zero is labelled for clarity.

community, a pattern emerges: lacking an α' value is effectively the same as an α' value of 0.00. Therefore, as α' increases the districts resemble the original structure more closely. It is also interesting to note that "patient zero", the first person infected in a given district, is at the end of the tail in all cases.

6.5 Conclusions and Future Work

The inclusion of districts with the model has allowed for the Graph Evolution Tool (GET) to increase its functionality when it comes to the goal of being able to use the model to simulate epidemics on human populations. However, there are several additional areas of development available to further improve the model.

6.5.1 Between-district and within-district infection rates

A logical starting point would be to further investigate the effect of modifying the values of α', the between-district infection rate and α, the within-district infection rate. In the current study α always had a fixed value of 0.5. Meanwhile, this study demonstrated that a lower value for α' had a positive effect upon epidemic duration. A more thorough investigation of the settings for these two parameters would provide greater insight on their impacts across the communities investigated here as well as other, larger, social contact networks. A further possibility is the implementation of α' as a function based on the number of infected individuals or some combination of properties within a community or its districts.

Chapter 7

Case Study 3

Modelling of Vaccination Strategies for Epidemics using Evolutionary Computation

This case study was originally published in CEC 2020 conference proceedings [22]. Personal contact networks that represent social interactions can be used to identify who can infect whom during the spread of an epidemic. The structure of a personal contact network has great impact upon both epidemic duration and the total number of infected individuals. A vaccine, with varying degrees of success, can reduce both the length and spread of an epidemic, but in the case of a limited supply of vaccine a vaccination strategy must be chosen, and this has a significant effect on epidemic behaviour.

In this study we consider four different vaccination strategies and compare their effects upon epidemic duration and spread. These are *random vaccination*, *high degree vaccination*, *ring vaccination*, and the base case of no vaccination. All vaccinations are applied as the epidemic progresses, as opposed to in advance. The strategies are initially applied to static personal contact networks that are known ahead of time. They are then applied to personal contact networks that are evolved as the vaccination strategy is applied.

7.1 Introduction

Recall that a *personal contact network* represents connections within a community of individuals along which a disease can spread. These networks can be generated through the use of demographic information, historical epidemic data, or by using an existing network such as YouTube watch history [40].

In the case of a limited supply of vaccine, selection of individuals for vaccination can have a significant impact on the severity of an epidemic. Four different vaccination strategies were considered in [50] and [44]. These included the simple strategy of choosing random individuals to be vaccinated, and three other strategies based on the structure of the personal contact network, which is known ahead of time.

The current study aims to evaluate four vaccination strategies to determine their relative effectiveness in reducing the length of an epidemic or cumulative number of infected individuals. Other than the baseline case in which no vaccination occurs, these strategies are *random vaccination*, *high-degree random vaccination*, and *ring vaccination*. The first two of these are also considered in [50] and [44] although we model the situation in which individuals are vaccinated *during* the time at which the epidemic is spreading. We measure the effect of the strategies upon epidemic duration and epidemic spread. Initially, the different strategies are applied to personal contact networks that are known ahead of time and static. The strategies are later applied to personal contact networks that evolve in reaction to vaccination, with these networks designed to maximize either epidemic duration or epidemic spread. We analyze the relative performance of the different strategies, as well as properties of the personal contact networks evolved.

7.2 Experimental Design

The Local THADS-N generative representation from Section 4.1 will be used in this case study to generate the personal contact networks which then are assessed for performance against the test problems.

In this study we consider the modelling of an epidemic with respect to two different fitness functions, each of which is combined with the four different vaccination strategies from Section 4.4. The two fitness functions are Epidemic Duration and Epidemic Spread from Section 4.3.

Table 7.1: Parameter Settings for probability densities of the edge operations. Created using a point packing with minimum distance of 0.535 from [21]. The header row is populated with PS meaning parameter setting, followed by the edge operations in the order listed in Section 5.2.1

PS	Togg.	Hop	Add	Del.	Swap	L-Togg.	L-Add	L-Del.	Null
1	0.0312	0.3528	0.2545	0.2835	0.0065	0.0151	0.0011	0.0324	0.0229
2	0.3988	0.0250	0.4167	0.0122	0.0380	0.0016	0.0325	0.0569	0.0182
3	0.0107	0.0358	0.7801	0.0243	0.0614	0.0563	0.0130	0.0057	0.0128
4	0.0438	0.0188	0.0281	0.0499	0.0003	0.4322	0.0051	0.0170	0.4047
5	0.0085	0.0104	0.0232	0.0177	0.4035	0.0008	0.0613	0.4326	0.0421
6	0.7592	0.0305	0.0024	0.0060	0.0068	0.0344	0.0106	0.1176	0.0326
7	0.0122	0.0196	0.3838	0.0942	0.0070	0.0067	0.0303	0.0173	0.4288
8	0.0020	0.0727	0.0010	0.0435	0.0355	0.0123	0.7946	0.0201	0.0184
9	0.0073	0.0261	0.3406	0.0015	0.0195	0.3741	0.0077	0.2037	0.0195
10	0.4303	0.0847	0.0105	0.0086	0.4158	0.0042	0.0025	0.0208	0.0226
11	0.0108	0.0072	0.0030	0.0617	0.4359	0.0010	0.0129	0.0223	0.4451
12	0.3682	0.0191	0.0319	0.3670	0.0191	0.1107	0.0127	0.0466	0.0247
13	0.0199	0.0032	0.3243	0.1881	0.3991	0.0012	0.0262	0.0269	0.0110
14	0.0180	0.0491	0.0002	0.7646	0.0277	0.0174	0.0661	0.0047	0.0522
15	0.0334	0.4149	0.0113	0.0005	0.0323	0.4130	0.0213	0.0373	0.0360
16	0.0129	0.3896	0.0193	0.0034	0.0038	0.0001	0.1483	0.4096	0.0130
17	0.0093	0.0211	0.0098	0.0020	0.8084	0.0532	0.0000	0.0519	0.0442
18	0.0341	0.0192	0.0044	0.2095	0.0586	0.3328	0.3233	0.0047	0.0134
19	0.0531	0.7894	0.0394	0.0261	0.0114	0.0203	0.0150	0.0164	0.0289
20	0.0038	0.0457	0.4066	0.0768	0.0226	0.0061	0.3952	0.0268	0.0164
21	0.0032	0.3982	0.0129	0.0426	0.0336	0.0115	0.0070	0.0539	0.4371
22	0.0259	0.0140	0.0032	0.2220	0.0056	0.0160	0.0203	0.3623	0.3308
23	0.0053	0.0229	0.0372	0.0192	0.0110	0.0134	0.0435	0.0045	0.8430
24	0.0176	0.3765	0.0125	0.0156	0.3637	0.0094	0.1757	0.0099	0.0190
25	0.3572	0.2016	0.0045	0.0079	0.0026	0.0088	0.3556	0.0510	0.0109
26	0.0027	0.0091	0.0011	0.0628	0.0071	0.8545	0.0484	0.0028	0.0115
27	0.0038	0.0156	0.0221	0.0021	0.0285	0.0267	0.4183	0.0396	0.4432
28	0.0265	0.0456	0.0144	0.0109	0.0179	0.0102	0.0087	0.7966	0.0691
29	0.4238	0.0343	0.0021	0.0319	0.0021	0.0066	0.0487	0.0037	0.4469

7.2.1 Static Graphs

The epidemic duration (ED) problem [8] seeks to find graphs which promote longer-lasting epidemics. In considering the epidemic duration problem, previous work [21] generated a large number of graphs. As a first step in the current study, we take the best 30 of these, i.e. those for which epidemics had the longest duration. Using these graphs, we then evaluate the length of the epidemic when each of the four vaccination strategies listed in Section 4.4 are applied.

The epidemic spread (ES) problem seeks to find graphs which promote more widespread epidemics, i.e. those in which more individuals are infected over the course of the epidemic. As this problem has not been previously addressed, we generate graphs designed to maximize epidemic spread, using the process defined in Section 4.2. Equivalently to the ED problem, we take the best 30 of these, i.e. those for which the maximum number of individuals were infected. We then evaluate the spread of the epidemic when each of the four vaccination strategies listed in Section 4.4 are applied.

Each of the above represents a situation in which a personal contact network is known ahead of time and does not change throughout the course of the epidemic, regardless of vaccine strategy applied.

7.2.2 Evolving Graphs

To represent a situation in which a personal contact network changes in reaction to vaccination, we also evolve graphs. To evolve the graphs, we follow the process defined in Section 4.2. Graphs are evolved with respect to (a) maximizing epidemic length and (b) maximizing epidemic spread. In each case, the graphs are evolved while a chosen vaccination strategy is being applied. As before, we use the four vaccination strategies described in Section 4.4.

7.3 Results and Discussion

This section provides the results for the performance of the four vaccination strategies. These are evaluated for their effect on both epidemic duration and epidemic spread, and for both static graphs and evolved graphs.

Note that the evolutionary algorithm is on the side of the epidemic, i.e. works towards providing personal contact networks which maximize the fitness (duration or spread). From the other side, the goal of the vaccination strategies is to reduce the

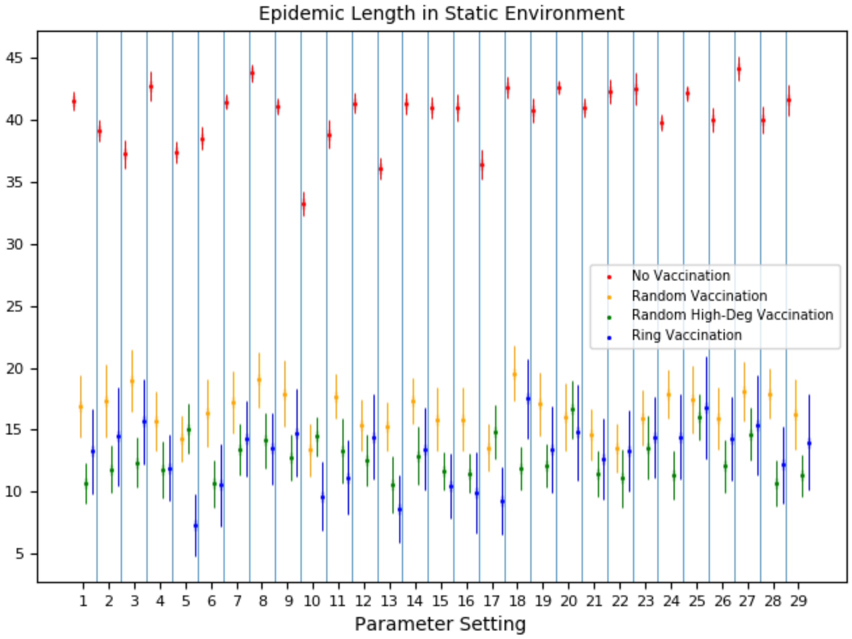


Figure 7.1: The mean epidemic length, with 95% confidence interval, of epidemics unleashed on the 30 best personal contact networks for each of the 29 parameter settings with the specified vaccination strategy. The personal contact networks are from [21].

length or spread of an epidemic. Therefore, in terms of vaccination strategy lower values are better, even though both fitness functions have the goal of maximizing their value during evolution.

7.3.1 Static Graphs

Epidemic Duration

The results for the 29 parameter settings under the epidemic duration problem, applied to a static environment with the four vaccination strategies, are available in Figure 7.1. Recall that the epidemic is not permitted to evolve in response to the chosen vaccination strategy in a static environment. Therefore, the epidemic length without any vaccination is expected to be much higher than with any vaccination strategy as the personal contact network was explicitly evolved to achieve a maximal epidemic length without any vaccinations. This turns out to be true: for the 29 parameter settings the mean epidemic length without vaccination ranged between 33.233 – 44.1667 compared to a drastically reduced 7.3 – 19.5667 for the other three vaccination strategies. Regardless of strategy, within a static environment the introduction of vaccines has a significant impact on overall length of the epidemic within

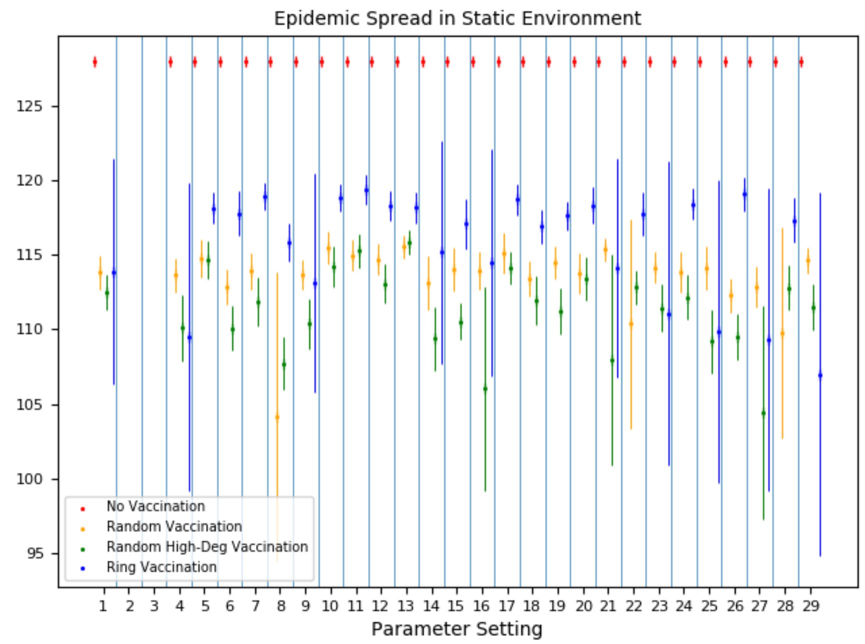


Figure 7.2: The mean epidemic spread, with 95% confidence interval, of epidemics unleashed on the 30 best personal contact networks for each of the 29 parameter settings with the specified vaccination strategy. The personal contact networks are from [21].

the population for all parameter settings.

Other than when there is no vaccination, the random vaccination strategy performs worst for all parameter settings except experiments 5, 17, and 20. However, the confidence intervals also typically overlap between these strategies and are thus not statistically significant. The random high-degree vaccination strategy performs exceptionally well in experiment 18 with a mean length of 11.9 ± 1.7319 compared to 17.5333 ± 3.195 for ring vaccination and 36.4333 ± 1.2142 without vaccination. Ring vaccination and high-degree vaccination perform similarly for the majority of experiments, though in experiments 5, 10, and 17 ring vaccination outperforms the other strategies. Lastly, the standard deviation for ring vaccination is larger than the other vaccination strategies leading to more variability in outcome when that strategy is chosen.

Epidemic Spread

The results for the 29 parameter settings under the epidemic spread problem, applied to a static environment with the four vaccination strategies, are available in Figure 7.2. The lack of a vaccine results in the totality of the population becoming infected during the course of an epidemic for each of the 30 runs performed on all parameter settings except for 2, 3, and 6. In experiments 2 and 3 the number of edges in the network happened to be higher than the cutoff to make the fitness evaluate to zero for all vaccination strategies. Recall that the epidemic spread fitness is zero whenever the total number of edges is more than 5 times the total number of nodes for a personal contact network. Also, in experiment 6, this was also the case for 16 of the 30 networks tested. These 16 fitness values of zero were removed from experiment 6 for all strategies before the mean fitness was calculated in order to not dramatically skew the results towards zero.

The inclusion of vaccination does an adequate job at limiting the spread of the epidemic throughout the population, regardless of which strategy is chosen. First, the random vaccination strategy does not outperform the other strategies nor does it under-perform them. Rather, the random strategy typically falls between the performance of high-degree and ring vaccination with the confidence intervals overlapping or one being contained within another. However, random vaccination can achieve superior fitness, though these scenarios greatly increase the variance of the fitness as can be seen in parameter settings 8, 22, and 28. Similarly, in experiments 16, 21, and 27 the random high-degree vaccination strategy also achieves fitness values that are less than the other strategies; this is at the cost of consistency of outcome when the strategy is applied. Finally, the ring vaccination strategy behaves in two distinct ways: a large variance and thus inconclusive results (e.g. for experiments 4 and 25) or a small variance and poor performance (e.g. experiments 11 and 26).

7.3.2 Evolving Graphs

Epidemic Duration

The results for the 29 parameter settings under the epidemic duration problem, applied to an evolving environment with the four vaccination strategies, are available in Figure 7.3. Recall that in an evolving environment a strategy for vaccination is chosen and the evolutionary algorithm is then permitted to find personal contact networks which maximize fitness while vaccines are present. Therefore, it is not surprising that the lengths achieved in environments with vaccine present perform much better than the static results from above. Most obviously, the ring vaccination strategy performs poorly compared to the random vaccination strategies. This could be because of the less random selection of individuals to vaccinate, permitting the evolutionary algorithm to develop strategies against the fixed strategy employed by ring vaccination.

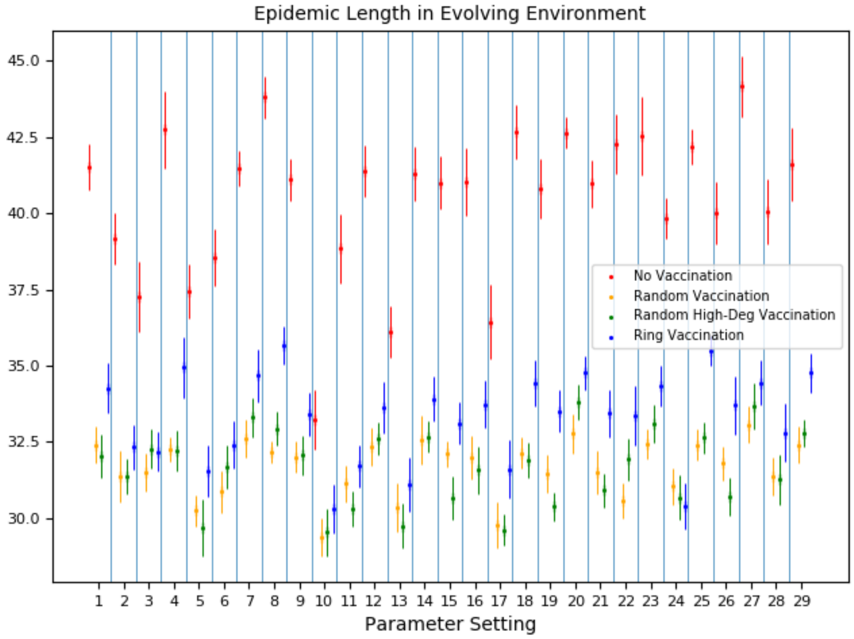


Figure 7.3: The mean epidemic length, with 95% confidence interval, of the best fitness value achieved on 30 runs of the evolutionary algorithm for each of the 29 parameter settings with the specified vaccination strategy.

In comparison, an entirely random strategy leaves little intuition to be gained by the evolutionary algorithm across generations.

The random vaccination strategies perform similarly to one another for all parameter sets, with the confidence intervals overlapping for all but experiments 15, 19, and 22. Random high-degree vaccination performs best in experiments 15 and 19 while the fully random vaccination strategy performs best in experiment 22. Furthermore, the overall reduction in the length of an epidemic is not consistent across parameter settings. For example, in experiment 10 the mean epidemic length went from 33.233 without a vaccine to 29.3667, 29.5333, and 30.3 for random vaccines, random high-degree vaccines, and ring vaccines respectively; this is a reduction of around three time steps. In contrast, in parameter setting 27 the mean lengths go from 44.1667 to 33.0667, 33.6667, and 34.4333, a reduction of around 10 time steps. Therefore, the overall impact of a vaccine on epidemic length is more dependent on the underlying personal contact network than the vaccination strategy being applied in this case.

Epidemic Spread

The results for the 29 parameter settings under the epidemic spread problem, applied to an evolving environment with the four vaccination strategies, are available in Figure

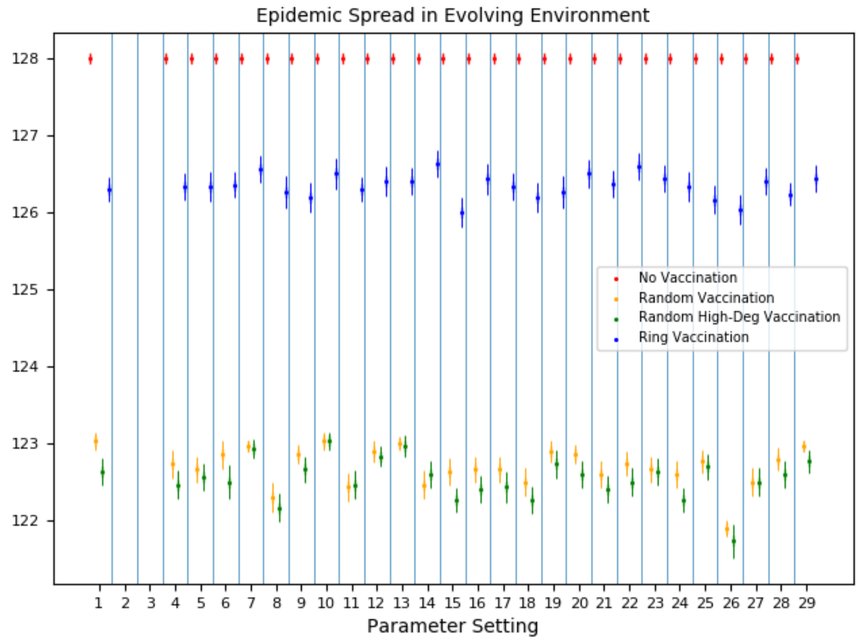


Figure 7.4: The mean epidemic spread, with 95% confidence interval, of the best fitness value achieved on 30 runs of the evolutionary algorithm for each of the 29 parameter settings with the specified vaccination strategy.

7.4. As with the static epidemic spread, parameter settings 2 and 3 both universally failed the fitness functions requirement that the total number of edges cannot exceed 5 times the total number of vertices, and are thus omitted to better see the contrast between the more interesting results from the other experiments. Also, experiment 6 once again has 16 of the 30 runs fail this requirement for all four vaccination strategies and these were removed before calculating the mean for experiment 6.

The ring vaccination strategy performs worse than both random strategies by a significant margin for all the experiments. Once again, this is likely a product of the evolutionary algorithm devising contact networks which counteract the effect of the ring vaccination strategy. Both random and high-degree strategies' results once again overlap in all but a few experiments, namely experiments 1, 15, and 24. The random high-degree strategy outperforms the random strategy for these parameter settings making them the optimal strategy in this case.

7.3.3 Static vs Evolving Graphs

The static environment does not allow for the personal contact network to respond to the vaccination strategy being applied to the population. Therefore, the outcomes of the strategies within the static environment have a much larger variance compared to those who undergo evolution. Thus, the result of applying the strategy varies greatly between runs. Whereas, when the networks are permitted to evolve the fitness values increase and variance decreases. This phenomenon is also what leads to the vaccine strategies performing better in a static environment as compared to one which is evolving.

7.3.4 Parameters

The parameter settings have a great deal of influence over the final networks being generated and thus they also greatly impact the performance of the various vaccine strategies above. Recall, experiment 2 and 3 failed when tested against the epidemic spread fitness function. Both of these settings had an abundance of *add* density with 0.4167 and 0.7801 respectively, which would result in an abundance of edges to be added to the graph likely leading to the epidemic spread fitness to evaluate to zero for all of the runs. Parameter setting 2 also had the *toggle* density set to 0.3988. Experiment 5 had a value of 0.4035 for *swap* and 0.4326 for *local delete*. *Local delete* would go about removing edges from triples of vertices which remain connected after the edge is deleted. This reduction in local connections can lead to a more restricted path for the epidemic to spread, improving the likelihood that the ring vaccination strategy immunizes a node necessary to access subsections of the graph. Experiment 5 accomplishes this on the epidemic length problem in a static environment. Parameter setting 6 also fails for 16 of the 30 runs when epidemic spread fitness is used which is likely the result of the density for *toggle* being 0.7592. As *toggle* can add or remove edges and the edges chosen are at random, it seems reasonable that this reliance on *toggle* is the culprit for why the fitness calculation fails approximately half the time.

7.4 Conclusions and Future Work

Four different vaccination strategies were evaluated for their effect upon epidemic duration and epidemic spread. These were first applied to personal contact networks known ahead of time, and that had been created specifically to maximize the duration or spread of the epidemic. They were then applied to personal contact networks that were evolved with a goal of maximizing epidemic duration or spread *as the vaccination strategy was applied.* For all of the above, the vaccinations were applied as the epidemic progressed, as opposed to in advance. This is closer to a real-life situation than one in which all vaccines are applied prior to the start of the epidemic and on a known, static graph.

When applied to a static network, it was demonstrated that all vaccination strategies greatly reduced both duration and spread of the epidemic. Random vaccination was tied to increased epidemic duration in comparison to both ring vaccination and high-degree vaccination, but generally slightly reduced spread in comparison to the others.

When applied to evolved networks, again all vaccination strategies reduced both duration and spread of the epidemic, but not as drastically. For these networks, random vaccination actually reduced epidemic duration and spread. This was possibly due to the fact that the evolutionary algorithm was unable to take advantage of a fixed strategy, and also the fact that the random vaccination strategy is more likely to choose a node further away from currently infected individuals, thereby allowing some sections of the graph to prevent infection passing through.

All of the above trends concerning relative performance of the vaccination strategies hold across all 29 parameter sets considered, with some minor variation. This is despite the fact that the graphs produced by the different parameter sets are quite different from one another.

Chapter 8

Case Study 4

Evolving the Curve

This case study was originally published in CIBCB 2020 conference proceedings [23]. Evolutionary algorithms are used to generate personal contact networks, modelling human populations, that are most likely to match a given epidemic profile. The Susceptible-Infected-Removed (SIR) model is used and also expanded upon to allow for an extended period of infection, termed the SIIR model. The networks generated for each of these models are thoroughly evaluated for their ability to match nine different epidemic profiles.

8.1 Introduction

This paper focuses on the ability to generate personal contact networks, representing physical connections between community members, which satisfy the data about the number of infections per time period. A personal contact network is the foundation of an epidemic model in which an epidemic spreads along the links of the network. This study compares two models of disease spread, in preparation for incorporating an asymptomatic state to match the behavior of SARS-Covid-2. The approach of employing a generative solution to a test problem is known as *graph induction* which has a variety of applications [16, 29, 34]. The representation used within this paper is known as the Local THADS-N generative representation; the metric used to evaluate the performance of a network is epidemic *profile matching*, introduced in [11]. The representation is described in detail in Section 4.1.

8.2 Background

8.2.1 The Models of Infection Used

The Susceptible-Infected-Removed (SIR) model of infection [30, 33] provides a simple model for the simulation of epidemics. In the SIR model, the epidemic disease lasts a single time step within an infected individual. The current study also allows for the infected stage to last two time steps, which we term *SIIR*; conceptually, this relates to a situation in which an individual is contagious for a longer length of time, thereby providing them an additional timestep in which they can infect others. This paper compares graphs evolved to match epidemic profiles with the SIR and SIIR models to assess the degree of influence the model has on the graph that arises. Another reason the SIIR model was chosen was in preparation for incorporating the *SEIR* model in which *Exposed* individuals have contracted the disease yet are not infectious; this is akin to the incubation period of a virus.

8.3 Experimental Design

A steady state evolutionary algorithm [43] is used to generate the solutions, which are strings of edge operations. All variables with respect to system design were determined empirically.

Table 8.1: The sets of Local THADS-N edge operation probabilities from [21] for constructing solutions using the evolutionary algorithm described in Section 4.2

Experiment	Toggle	Hop	Add	Delete	Swap	L-Toggle	L-Add	L-Delete	Null
PS1	0.2528	0.0142	0.0087	0.2138	0.0021	0.2267	0.2228	0.0079	0.0509
PS2	0.0056	0.0016	0.0133	0.0032	0.0272	0.0177	0.8115	0.0214	0.0985
PS3	0.2713	0.0041	0.0133	0.0129	0.0311	0.0446	0.5675	0.0068	0.0484
PS4	0.0044	0.0419	0.0082	0.0149	0.0135	0.3141	0.5054	0.0366	0.0611
PS5	0.0090	0.0002	0.3233	0.0183	0.0020	0.0128	0.4968	0.0249	0.1128
PS6	0.4925	0.0061	0.0120	0.0517	0.0083	0.0052	0.2846	0.0127	0.1268
PS7	0.0197	0.0795	0.7238	0.0001	0.0481	0.0175	0.0393	0.0038	0.0684
PS8	0.0084	0.0159	0.0323	0.0172	0.0021	0.0046	0.4702	0.3775	0.0718

8.3.1 An Entropic Pseudometric for Comparing Graphs

One of the major hypotheses under test in this study is that, within the same epidemic profile, changing the model of disease spread will cause the graph induction system to produce substantially different graphs. To document this, we need a way to compare graphs that is not obfuscated by the many irrelevant variations in structure. A pseudometric that has these qualities is the Column-Entropy distance(CE-distance) [35]. Computation is based on the simulated diffusion of a collection of different gasses, one per vertex, with absorption of all gasses present taking place at a low rate at each vertex. This process converges rapidly as gas is added arithmetically but decays exponentially via absorption. The result is a matrix, with rows indexed by vertices and columns indexed by gasses, giving the amount of each gas at each vertex. Columns are normalized to sum to one and then the entropy of each column is computed, yielding an entropy vector for the network. The entropy associated with a column represents the evenness of distribution of other nodes as destinations of random walks beginning at the node indexing the column. Entropies are then sorted into decreasing order to create sorted entropy vector for a network. The CE-distance between two networks is the Euclidean distance between their sorted entropy vectors. The sorting step is a fast method of approximating correspondence between nodes in the two networks. A more detailed explanation of this, and other pseudometrics on networks, appears in [35]. A pseudometric is a distance measure with the property that two dissimilar objects can be at distance zero from one another – something that did not occur in practice in this study, meaning that the CE-distance is functionally a metric in this study.

8.4 Results and Discussion

Taking the network with best (lowest) fitness from each run of the evolutionary algorithm results in 30 graphs for each (parameter setting, profile) pair. These were used to generate box and whisker plots of the profile matching fitness on the nine profiles using the SIIR model of infection, shown in Figure 8.1. Additionally, results from previous work using the SIR model of infection are included [21] to investigate the impact of increasing the infectious period of an epidemic. It is clear that the fitness values achieved remain consistent between the SIR and SIIR epidemic models. The confidence intervals achieved by both models overlap, with the parameter settings having similar impacts on performance regardless of the model chosen. This demonstrates that the addition of the SIIR model does not have a tangible impact on the overall fitness of the networks generated. However, although the fitness values are similar other differences can exist within the networks, which will be investigated in the following sections.

Table 8.2: The parameter setting with the lowest (best) mean fitness across 30 runs for each profile and model of infection used.

Profile	SIR Best	SIR Mean	SIIR Best	SIIR Mean
1	PS3	7.7314	PS2	7.7023
2	PS3	9.8480	PS3	9.8514
3	PS3	8.7472	PS3	8.7454
4	PS3	7.7277	PS3	7.7803
5	PS2	9.6765	PS2	9.5650
6	PS2	8.5336	PS2	8.4512
7	PS2	9.9484	PS2	9.8524
8	PS7	7.1829	PS7	7.1637
9	PS2	7.3203	PS2	7.3143

8.4.1 Graph Visualizations

To investigate the impact of the SIIR model on the networks generated, visualizations of the graphs were created. The network with best (lowest) fitness from the parameter setting with the lowest mean fitness is chosen from each of the epidemic models studied. See Table 8.2 for the parameter settings and mean fitness corresponding to the visualizations being compared. To aid in analyzing the differences between networks the nodes were coloured as follows: patient zero is red, nodes 1-31 are cyan, nodes 32-63 are orange, nodes 64-95 are yellow, and nodes 96-127 are green. Visuals

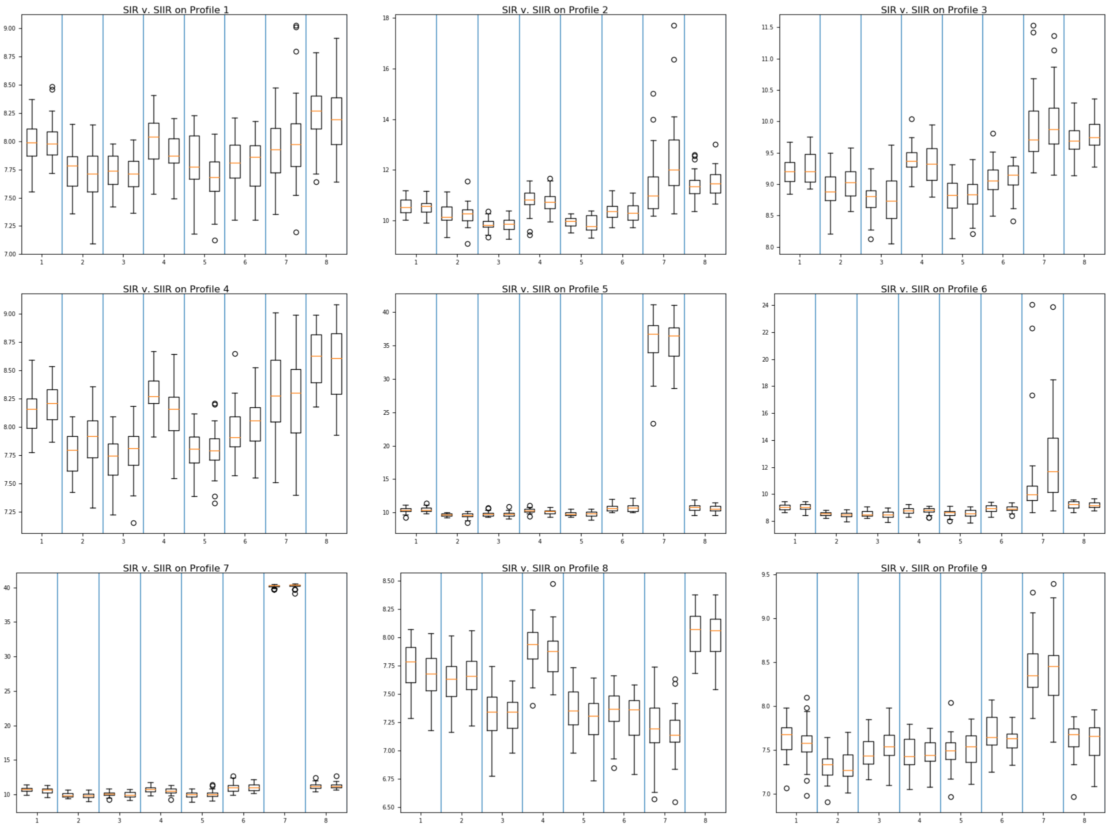


Figure 8.1: Box and whisker plots of the profile matching fitness achieved on 30 runs of the evolutionary algorithm using the SIR and SIIR epidemic models. The eight columns correspond to the eight parameter settings with the left box plotting the fitness using the SIR model, and the right the SIIR model within each column.

for profiles 2, 4, and 6 are provided respectively in Figures 8.2, 8.3, and 8.4.

Looking at these visualizations there seem to be clear differences between the networks on the two models of infection. For profile 2 the SIR model results in significant intermixing of the four node colours throughout the network. The SIR model also retains a lot of the chain in the initial network from Figure 4.1, most notably with the green nodes at the top left and cyan nodes at the bottom right. In contrast the SIIR model results in much less intermingling of the four colours with four distinct clusters remaining intact in the evolved network while much of the chain structure is lost.

The retained chain structure is a feature of both networks generated on profile 4 in Figure 8.3, although the location of the chain within the resultant network is different between models. The SIR model contains distinct chains in the blue and orange nodes, while the SIIR model features chains in the yellow and orange nodes. Flipping one of the networks reveals that the overall structure of the network is similar, with the non-chain section of the network resembling a large well connected cluster of all node colours and patient zero.

The networks generated using profile 6 in Figure 8.4 feature large sections of the initial structure of the network. The SIR model is made almost entirely of the chain structures with most of the orange, yellow and green nodes being part of the chain. The SIIR network also has large sections of the chain retained for green, blue, and orange nodes, although less notably than in the SIR model. Also, the SIIR model allows for more mingling between nodes of different colours than the SIR model.

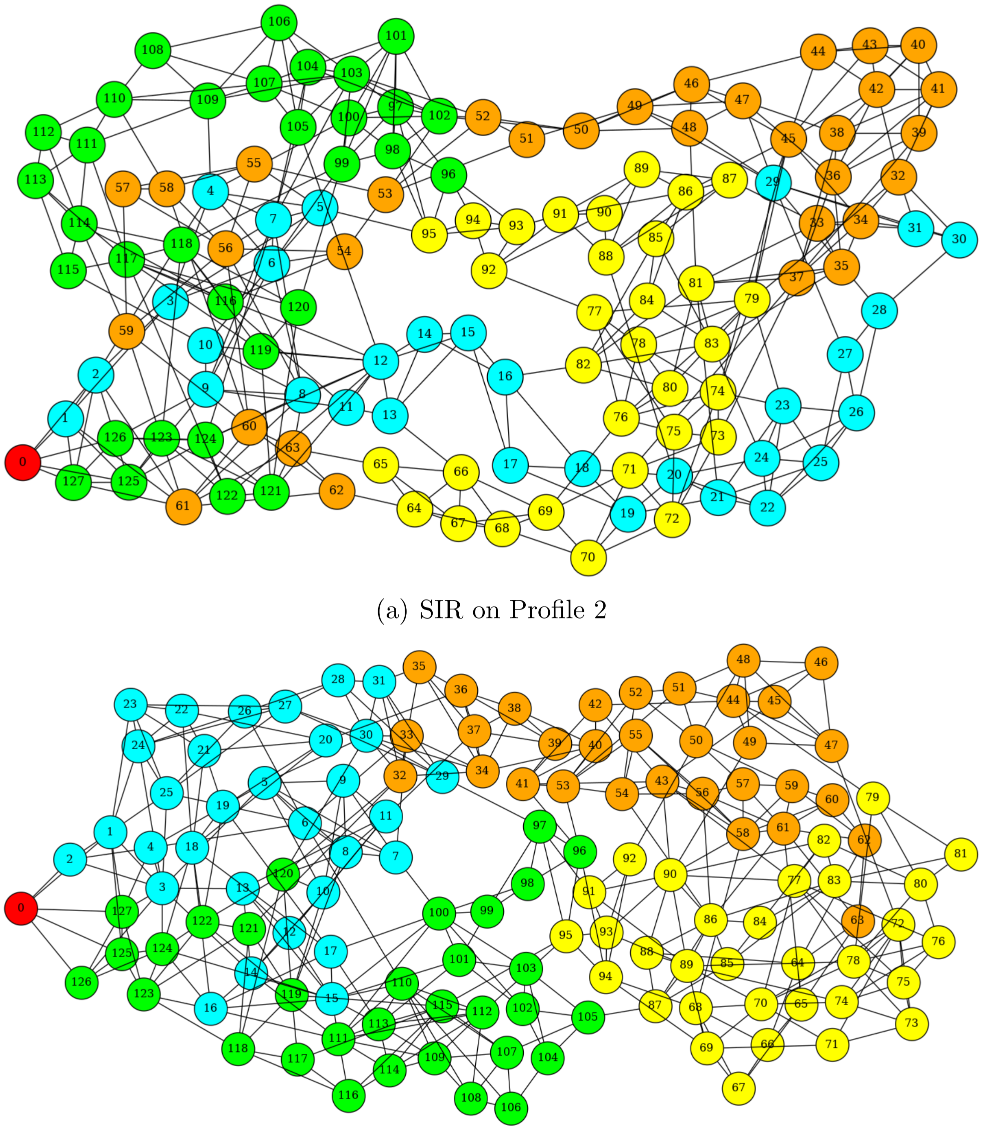
The differences between networks generated using different models of infection are similar for the remaining profiles, although the changes are not as stark as for the networks from the above profiles. A more thorough investigation is necessary to gain insight into the patterns that exist on the plethora of networks generated in this study.

8.4.2 An Entropic Pseudometric for Comparing Graphs

The final structure of the networks generated can fluctuate based on the various parameter settings, profiles and epidemic models used within the study. Therefore the column entropy distance is computed in a pair-wise manner to determine the distance between any pair of networks. The network used to compute this value for any given system configuration is the network which achieved the best fitness across the runs. These values were used to generate heat maps of the column entropy distance between and within the two epidemic models; see Figure 8.5. Dark blue represents a difference of zero while bright yellow indicates a high distance.

The first heat map compares the 72 networks generated using the SIR model with the 72 generated using the SIIR model. Different patterns emerge across system configurations. The top-left quadrant demonstrates that the networks are somewhat impacted by the epidemic model being used, with some cells being darker blue and others approaching the median value. In contrast the center, specifically comparing profile 5 to 7 across models, features networks with minimal column entropy distance. Furthermore, profile 8 and 9 are the brightest rows/columns revealing the large variability between the networks realized under the two models. Most notably, profile 9 under SIR results in networks furthest from those generated using SIIR on profile 8, with this section of the map being brightest.

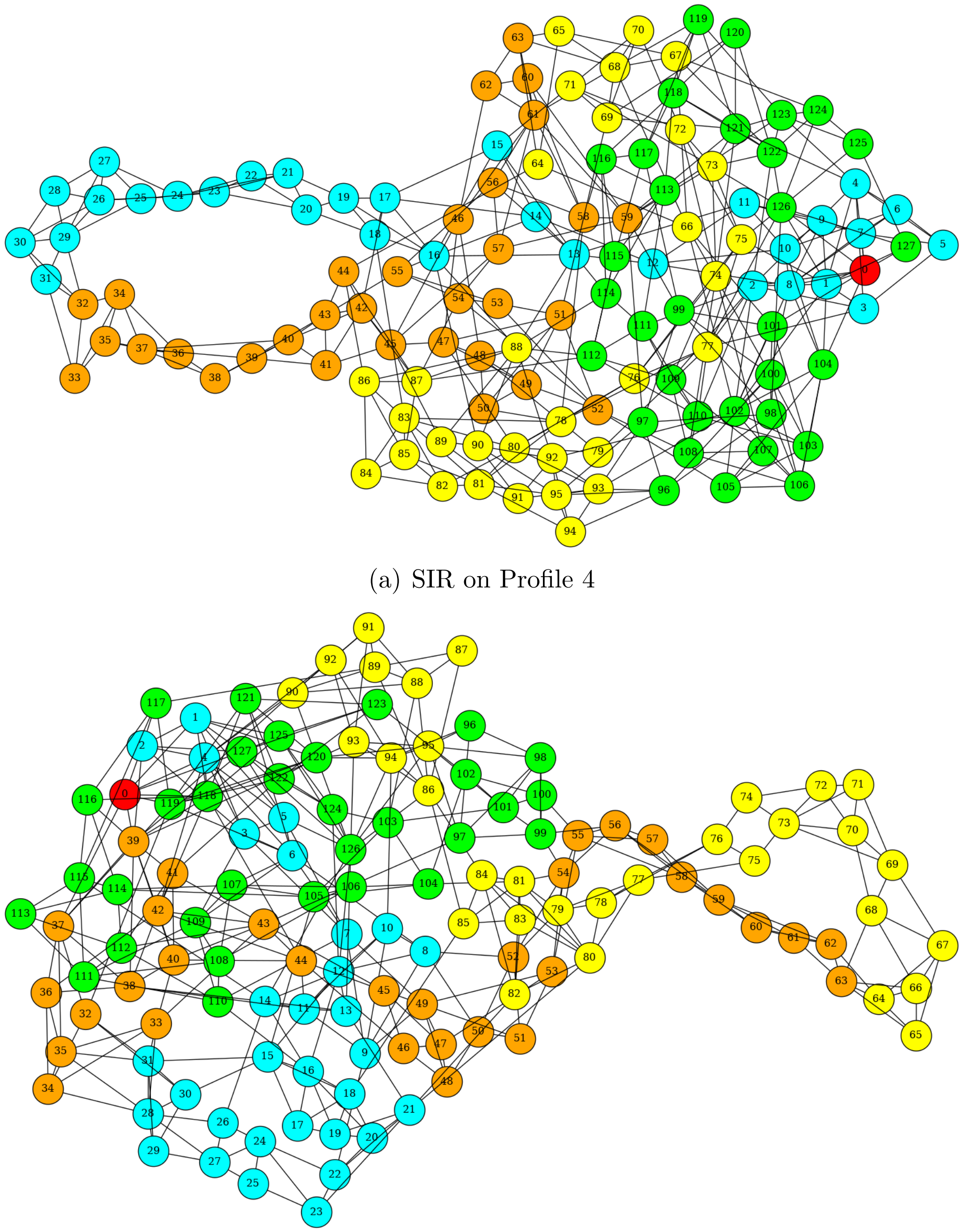
The second heat map compares networks generated using the SIR model with themselves, and the third map does this for the SIIR model. The patterns from the



(a) SIR on Profile 2

(b) SIIR on Profile 2

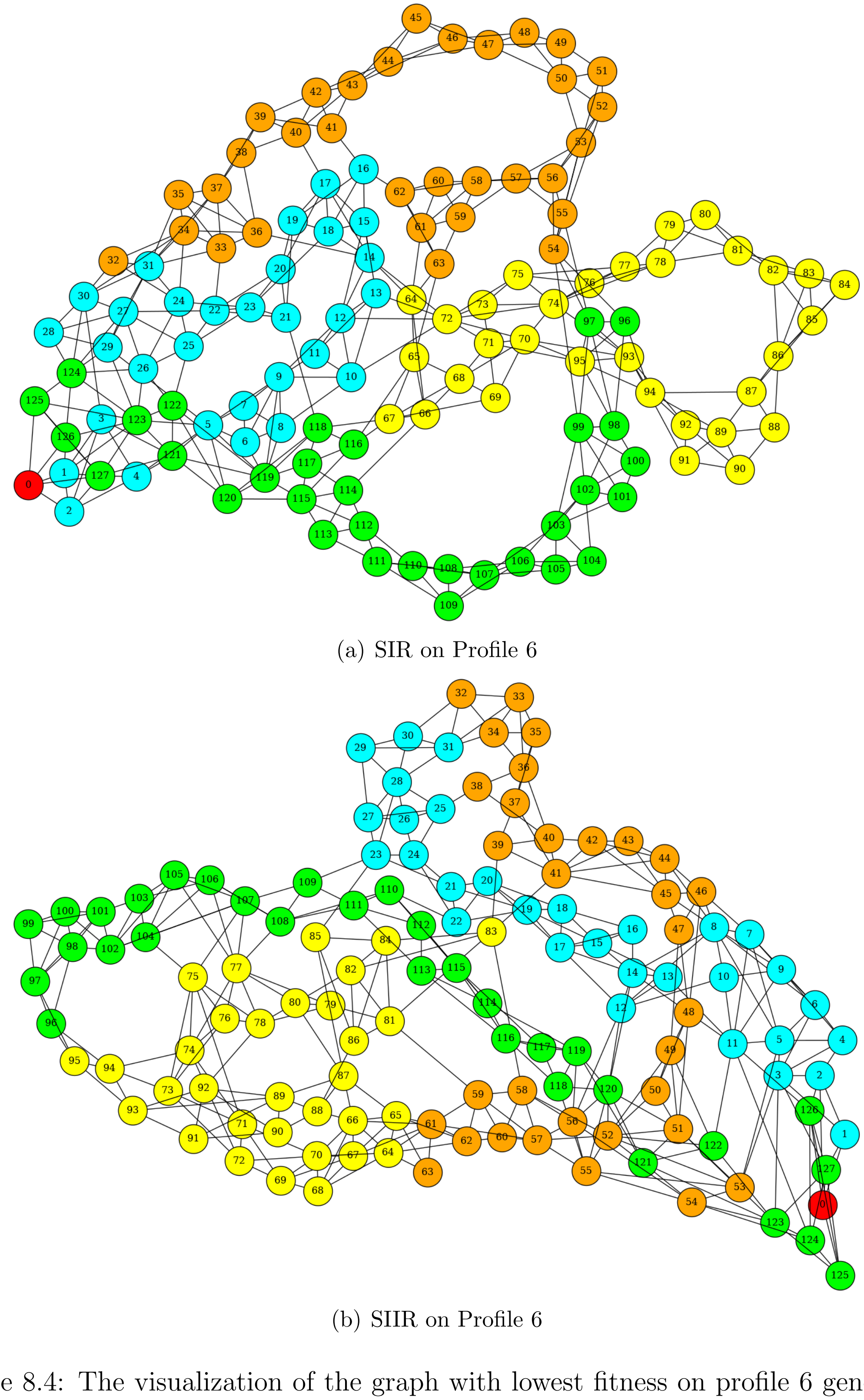
Figure 8.2: The visualization of the graph with lowest fitness on profile 2 generated under the specified model of infection. See Table 8.2 for parameter setting and fitness values.



(a) SIR on Profile 4

(b) SIIR on Profile 4

Figure 8.3: The visualization of the graph with lowest fitness on profile 4 generated under the specified model of infection. See Table 8.2 for parameter setting and fitness values.



(a) SIR on Profile 6

(b) SIIR on Profile 6

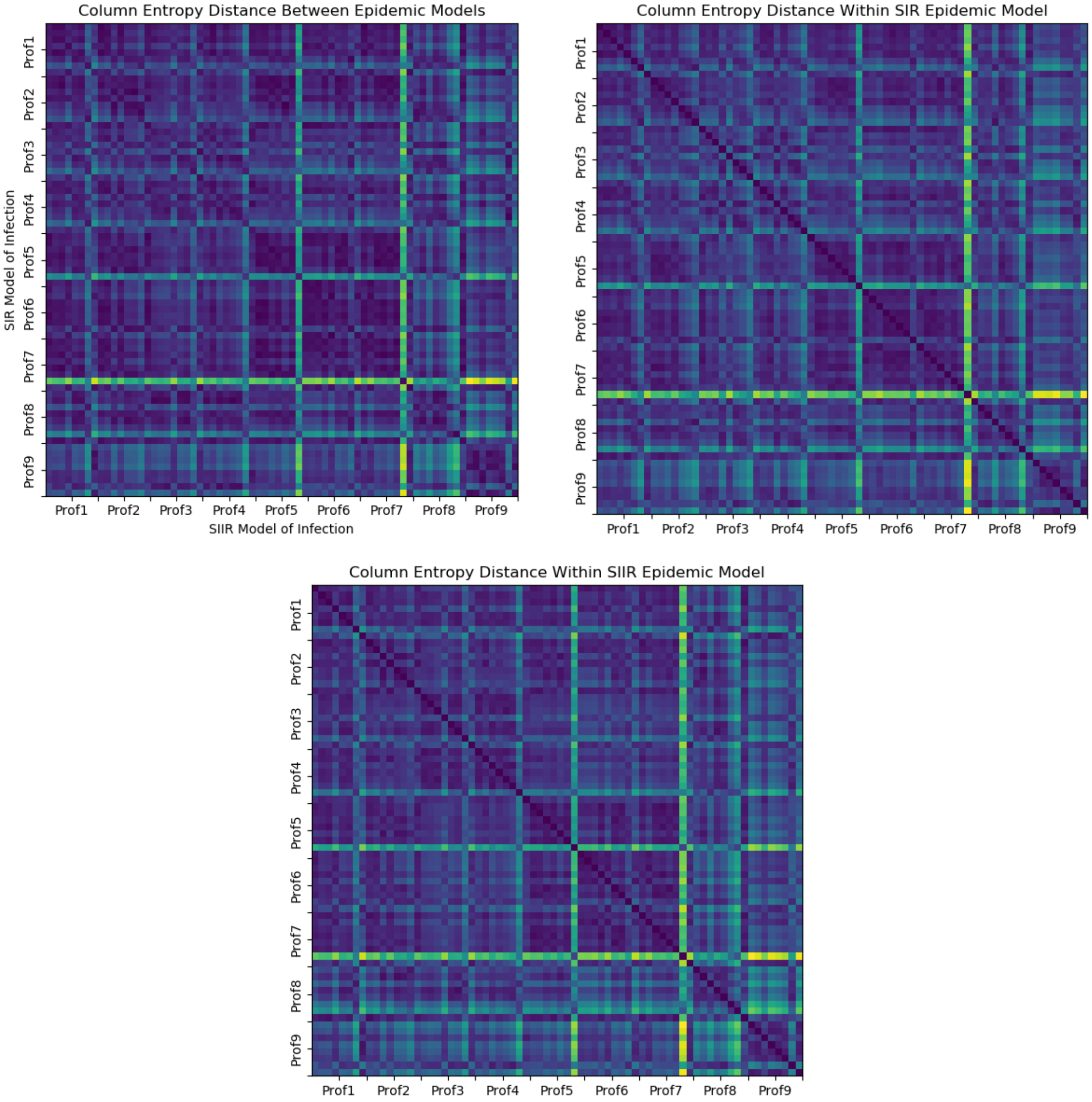
Figure 8.4: The visualization of the graph with lowest fitness on profile 6 generated under the specified model of infection. See Table 8.2 for parameter setting and fitness values.

first heat map are largely apparent in these maps as well with only slight variations. This provides evidence that the networks generated are dominated by the profile and to a lesser degree the parameter setting being used to select edge operations. Therefore, the variable having the least impact is in fact the model of infection chosen, although small variations exist between the maps. Under the SIR model the top-left quadrant demonstrates less variability between networks from settings on profiles 1-4. The center of the map is noticeably brighter than the first figure, meaning that the distance between networks generated within the SIR environment actually differ *more* than those between the models, an unexpected result.

The final heat map is the brightest of the three. This means that the networks created using the SIIR epidemic actually differ more from each other than from those generated using the SIR model. Once again the most significant variations exist for the networks created using profile 8 and 9. These profiles are comprised of epidemic curves that are heavily weighted towards the start of an outbreak. This would favour sporadic epidemic behaviour in which the outcome of the epidemic depends heavily on where and how quickly the virus spreads in the first few time steps. This likely contributes to the variability between networks observed in the heat map. The rows and columns that are coloured almost entirely yellow and green in profile 5 and 7 under PS7 are due to the reliance on the add operation within that parameter setting. This causes the epidemics to quickly spread and infect the entirety of a population early on. However, these profiles feature a significant portion of their infections later in the outbreak so the fitness is impacted significantly, leading to the large column entropy distance.

8.4.3 SIR on SIIR, etc. Epidemic Profiles

The final method of comparison between the epidemic models involves the epidemic profiles or epidemic curve that result from simulating an epidemic. The network with the lowest mean fitness, across all parameter settings, is used for each epidemic model. This provides two networks per epidemic profile. On each of the networks 500 SIR and 500 SIIR epidemics are simulated; the SIR and SIIR epidemic with lowest fitness for each model is then plotted against the profile being evolved to. This plot is available for profile 1 in Figure 8.6. The networks generated using the SIR model result in curves which most closely resemble the profile being evolved to, irrespective of the type of epidemic being actualized. Although, the network that came from the SIIR model results in two epidemics which overshoot the epidemic curve but in


Column Entropy Distance Between Epidemic Models

Column Entropy Distance Within SIR Epidemic Model

Column Entropy Distance Within SIIR Epidemic Model

Figure 8.5: Heatmaps of column entropy distance between graphs generated using the specified epidemic profile and model of infection. Each profile consists of eight vectors representing results from each of the parameter settings on that profile.

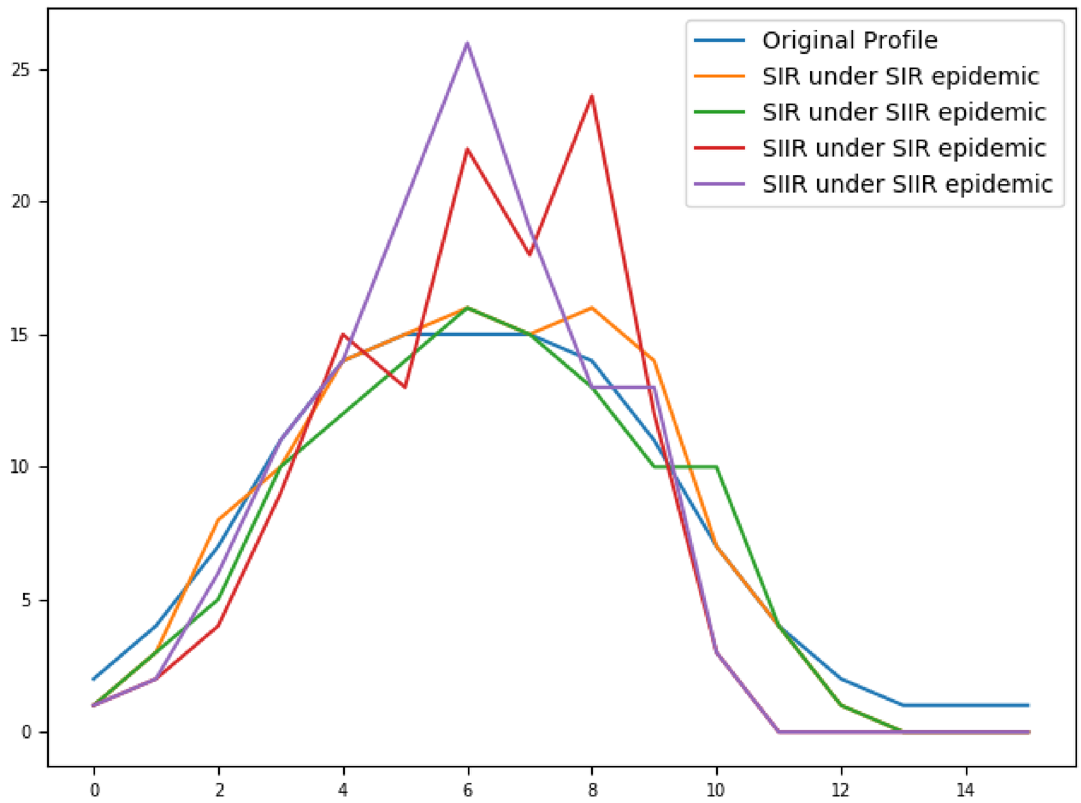


Figure 8.6: Number of infected individuals per time step for simulated epidemics on networks generated using the evolutionary algorithm from Section 5.2.1 using profile 1. The network with lowest mean fitness evolved using each epidemic model of infection was used to simulate 500 SIR and 500 SIIR epidemics. The epidemic with best fitness is shown.

a similar manner. The SIIR epidemic provides a steeper and taller curve than the SIR epidemic's more jagged and delayed peak. This is likely because the increased infection length allows for greater and faster spread of the epidemic.

8.5 Conclusions and Future Work

The addition of the SIIR model of infection was hypothesized to result in differences in the networks generated using the system described above. This addition did provide evidence that the model of infection has an impact on the networks generated though the differences are minor in the majority of cases. The algorithm was able to adapt to the SIIR model by matching the fitness achieved by the SIR model in all cases. The visualizations of the networks generated demonstrate that differences in structure are present based on the model used. This structure allows for two different epidemic models to generate the same epidemic curve on their respective network, as shown in Figure 8.6. Lastly, the column entropy distance heat maps revealed that the profile, parameter setting, as well as the epidemic model all contribute to the networks generated and fitness achieved, although the profile and parameter setting cause the majority of the fluctuation when compared to the model of infection being deployed.

Chapter 9

Conclusion

The Local THADS-N representation has many potential avenues for improvement in order to better model epidemics and generate the personal contact networks used to do so. These possibilities and the current progress in achieving them will be outlined in the proceeding sections.

9.1 Point Packing

The point packing method for selecting parameters continues to provide ample evidence as to how it consistently finds PSs which have a greater ability to explore more of the parameter space. This exploration has resulted in the best fitness values found on all of the epidemic profiles. Additionally, the point packing approach does not require anything other than for the researcher to set a minimum acceptable distance between points. These benefits come with a method for parameter selection that works to remove researcher bias or the need for background information on the problem. This is especially true when considering that the same parameter set can have drastically different results across environments. Without using a point packing it is likely that any set of parameters chosen in traditional ways will not explore enough of the parameter space.

9.1.1 Improved point packing

In this study we used point packings that were generated uniformly at random and then normalized to generate probabilities for the various graph editing operations that add to 1.0. It turns out that this method of finding point packings can be problematic, as it has a bias toward the probability $1/n$ where n is the number of parameters being set. A more complex form of normalization permits uniform sampling of the simplex of points with positive coordinates summing to 1.0. While the parameter setting performed here is not incorrect, a more complete sampling might be possible with the new technique.

The inclusion of the SIRS model is the first of many possibilities when it comes to exploring more complex epidemic models. More research should be conducted into potential patterns between a PSs ability to generate successful networks within various epidemic environments. Additionally, ways to reduce the volatility present within the longer lasting epidemics associated with the SIRS model should be developed. This relates closely to the continued goal of reducing the complexity of the evolutionary algorithm utilized within this paper to allow for better scalability.

Other modifications could include expansions to the epidemic models used here. For example, the graphs could use directed edges, or the probability of infection could be modelled by using weighted edges. Furthermore, the representation can be applied to new graph-evolution problems which may solve problems in new and interesting domains. Lastly, this model could be tested against real-world data sets from historical epidemics or outbreaks on campuses, in hospitals or between communities for which data has been recorded and is available. Endless possibilities exist for the expansion of this generative representation.

The use of point packing was included as a means of finding an optimal set of parameter settings for the problem at hand. In the future, hyper-parameter tuning could be examined [13]. To go one step further, we could also investigate the use of hyper-heuristics [14].

9.2 Epidemic Model and Community Properties

Previous work included the use of two epidemic models, the SIR model used here, and the SIRS model where an individual once again becomes susceptible some number of time steps after recovering from infection. It would be worthwhile to study the impact of the SIRS model when used with districts. Another modification could include the ability to represent voluntary vaccination rates based on the severity of the epidemic among other factors as presented in [38]. Additionally, it would be beneficial to thoroughly investigate the number of districts and their sizes, along with the possibility of connecting the districts in an adaptive manner, or at least not simply as a fully connected graph.

9.3 Algorithm Complexity

As the representation has grown and evolved, so too has the running time. A necessity of epidemics lasting longer is that the fitness calculations, which are a simulated epidemic, take longer to calculate. Therefore, as the model becomes more complex, complications will compound and further iteration, without simplification, will become increasingly challenging. Furthermore, large increases to community size would allow for the model to be used on larger human populations, although this would also dramatically impact the computation time.

9.4 Comparing Solutions and Graph Visualization

There exist other tools which can generate graphs to test against their ability to model epidemic behavior. Therefore, a future area of research could be to compare the Local THADS-N representation against other solutions. This work could include examining the success of these methodologies in modeling real world epidemic behavior. Furthermore, it would be interesting to examine the actual graphs generated, including possibly by using graph visualization tools in order to view epidemic behaviour on a community.

In the future, it would be of interest to consider how to evolve the vaccination strategies themselves, using known, static, personal contact networks. These evolved strategies could then be compared to existing strategies, including those presented in this paper. One could also consider such factors as a sliding scale of high-degree vaccination. It would also be of interest to apply these ideas and techniques to related issues such as the design of quarantine strategies.

Also, in from Chapter 7 current study we considered that vaccines would take effect immediately. In real life situations, this can depend upon the type of vaccine. Therefore we could consider the following vaccine effectiveness delays: (i) that the vaccine is effective immediately, (ii) that the vaccine takes one time step to become effective, and (iii) that the vaccine takes two (or more) time steps to become effective.

A key limitation of this system is the size of the personal contact networks that can be generated before evolution becomes too costly to be practical. Networks with 128 nodes can model small communities while simulating an epidemic between communities as part of a personal contact network resembling a city, province, or country. This would also allow for smoother epidemic curves as the network would more closely resemble real world networks with physical and social distance between members of the population, a situation that is not possible with 128 nodes. Moreover, the epidemic curves released by public health professionals often feature rolling averages to handle sporadic epidemic behaviour that changes day-by-day. Larger networks, permitting longer epidemics, would allow for smoother epidemic curves than those realized within this study.

The inclusion of the SIIR model is the first of many possibilities when it comes to exploring more complex epidemic models. A further increase to the length of infectability, such as in [4], along with the inclusion of a presymptomatic or asymptomatic state to the model are also possible. More research should be conducted into potential patterns between the ability of a parameter setting to generate successful networks within various epidemic environments. This relates closely to the continued goal of reducing the complexity of the evolutionary algorithm utilized within this paper to allow for better scalability.

Other modifications could include expansions to the epidemic models used here. For example, the graphs could use directed edges, or the probability of infection could be modelled by using weighted edges. Future work would also benefit from exploring different values for α or replacing a single value with a probability distribution to better resemble real-world virus infectability. Additionally, exploring different initial graphs and their impact on performance could provide new insights and increase the robustness of the software. The personal contact networks from particular countries or communities will undoubtedly have some variation and the software will need to handle this variation. Furthermore, the representation can be applied to new graph-evolution problems which may solve problems in new and interesting domains. Lastly, this model could be tested against real-world data sets from SARS-Covid-2, historical epidemics or outbreaks on campuses, in hospitals or between communities for which data has been recorded and is available. Endless possibilities exist for the expansion of this generative representation.

9.5 Investigating Different Initial Networks

The network that undergoes strings of edge operations from Section 4.1 is quite different from personal contact networks from real populations. It is also quite common for the final networks to retain part of the initial ring. Therefore the use of an initial network with a different structure, possibly resembling aspects seen in real world graphs, would give a greater opportunity for evolution to find more realistic networks.

Bibliography

[1] Faruque Ahmed. U.S. advisory committee on immunization practices handbook for developing evidence-based recommendations. Technical report, Centre for Disease Control and Prevention, 2013.

[2] Joan L. Aron and Ira B. Schwartz. Seasonality and period-doubling bifurcations in an epidemic model. *Journal of Theoretical Biology*, 110(4):665–679, 1984.

[3] Daniel Ashlock and Lee-Ann Barlow. A class of representations for evolving graphs. In *Proceedings of the 2015 Congress on Evolutionary Computation*, pages 1295–1302, 2015.

[4] Daniel Ashlock, Joseph Brown, and Clinton Innes. *Dawn of the Dead : An Evolvable Linear Representation for Simulating Government Policy in Zombie Outbreaks*, page 233–248. University of Ottawa Press, 10 2014.

[5] Daniel Ashlock and Steffen Graether. Conway crossover to create hyperdimensional point packings, with applications. In *Proceedings of the 2016 Congress on Evolutionary Computation*, pages 1570–1577, Piscataway, NJ, 2016. IEEE Press.

[6] Daniel Ashlock and Fatemeh Jafargholi. Evolving extremal epidemic networks. In *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pages 338–345, Piscataway NJ, 2007. IEEE Press.

[7] Daniel Ashlock and Fatemeh Jafargholi. Representations for the evolution of extremal epidemic networks. In *Proceedings of the 2008 World Congress on Computational Intelligence*, pages 660–667, Piscataway NJ, 2008. IEEE Press.

[8] Daniel Ashlock and Colin Lee. Characterization of extremal epidemic networks with diffusion characters. In *Proceedings of the 2008 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pages 264–271, Piscataway NJ, 2008. IEEE Press.

[9] Daniel Ashlock and Colin Lee. Diffusion characters: Breaking the spectral barrier. In *Proceedings of the 21st Canadian Conference on Electrical and Computer Engineering*, pages 847–850, Piscataway NJ, 2008. IEEE Press.

[10] Daniel Ashlock, Justin Schonfeld, Lee-Ann Barlow, and Colin Lee. Test problems and representations for graph evolution. In *Proceedings of the IEEE Symposium on the Foundations of Computational Intelligence*, pages 38–45, Piscataway NJ, 2014. IEEE Press.

[11] Daniel Ashlock and Elisabeth Shiller. Fitting contact networks to epidemic behavior with an evolutionary algorithm. In *Proceedings of the 2011 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pages 1–8, Piscataway NJ, 2011. IEEE Press.

[12] Daniel Ashlock and Meghan Timmins. Adding local edge mobility to graph evolution. In *Proceedings of the 2016 IEEE World Congress on Computational Intelligence*, Piscataway NJ, 2016. IEEE Press.

[13] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(10):281–305, 2012.

[14] Edmund Burke, Graham Kendall, Jim Newall, Emma Hart, Peter Ross, and Sonia Schulenburg. *Hyper-Heuristics: An Emerging Direction in Modern Search Technology*, chapter 16. 01 2003.

[15] Charles Darwin. *The Origin of Species*. Five foot shelf of books. P. F. Collier, 1909.

[16] Alexandre Devert, Thomas Weise, and Ke Tang. A study on scalable representations for evolutionary optimization of ground structures. *Evolutionary computation*, 20(3):453–472, 2012.

[17] Richard C. Dicker, Fátima Coronado, Denise Koo, and Roy Gibson Parrish. *Principles of epidemiology in public health practice: an introduction to applied epidemiology and biostatistics*, pages 13–108. U.S. Department of Health and Human Services, Centers for Disease Control and Prevention (CDC), Office of Workforce and Career Development (OWCD), Career Development Division (CDD), Atlanta, Georgia 30333, 2012.

[18] The Lancet Infectious Diseases. The imperative of vaccination. *Lancet Infectious Diseases*, 17(11):1099, Nov 2017.

[19] Michael Dubé. The local thads-n generative representation for epidemic profile matching. Undergraduate thesis, Brock University, 05 2018.

[20] Michael Dubé, Sheridan Houghten, and Daniel Ashlock. Parameter selection for modeling of epidemic networks. In *Proceedings of the 2018 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, Piscataway NJ, 2018. IEEE Press.

[21] Michael Dubé, Sheridan Houghten, and Daniel Ashlock. Representation for evolution of epidemic models. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, pages 2370–2377, Piscataway NJ, 06 2019. IEEE Press.

[22] Michael Dubé, Sheridan Houghten, and Daniel Ashlock. Modelling of vaccination strategies for epidemics using evolutionary computation. In *2020 IEEE Congress on Evolutionary Computation*. IEEE, 2020.

[23] Michael Dubé, Sheridan Houghten, Daniel Ashlock, and James Hughes. Evolving the curve. In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–8, 2020.

[24] Michael Dubé, Sheridan Houghten, and Daniel Ashlock. Pandemic: A graph evolution story. In *2019 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–8, Piscataway NJ, 2019. IEEE Press.

[25] Douglas J. Futuyma and Mark Kirkpatrick. *Evolution*. Sinauer Associates, 2017.

[26] Yuanzheng Ge, Zhichao Song, Xiaogang Qiu, Hongbin Song, and Yong Wang. Modular and hierarchical structure of social contact networks. *Physica A Statistical Mechanics and its Applications*, 392:4619–4628, 10 2013.

[27] Harry B. Greenberg and Mary K. Estes. Rotaviruses: From Pathogenesis to Vaccination. *Gastroenterology*, 136(6):1939–1951, May 2009.

[28] Frederic Guinand, Patrick Siarry, and Nicolas Monmarché. *Artificial Ants: from collective intelligence to real-life optimization and beyond*, pages 19–44. Wiley, 01 2010.

[29] Philip Hingston, Luigi Barone, and Zbigniew Michalewicz. *Design by Evolution: Advances in Evolutionary Design*. Springer, New York, NY, 2008.

[30] Frank Hoppensteadt and Charles Peskin. *Mathematics in medicine and the life sciences*, volume 10. Springer Science & Business Media, 2013.

[31] Greg Hornby. Improving the scalability of generative representations for openended design. In R. Riolo, T. Soule, and B. Worzel, editors, *Genetic Programming Theory and Practice V*, Genetic and Evolutionary Computation Series, pages 125–142. Springer, Boston, MA, 2008.

[32] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948, Piscataway NJ, 1995. IEEE Press.

[33] William Kermack and Anderson McKendrick. Contributions to the mathematical theory of epidemics. *Bulletin Mathematical Biology*, 53(1):33–55, Mar 1991.

[34] Taras Kowaliw, Nicholas Bredeche, and René Doursat. *Growing Adaptive Machines*. Springer, 2014.

[35] Colin Lee and Daniel Ashlock. Diffusion characters: Breaking the spectral barrier. In *Proceedings of the 2008 Canadian Conference on Electrical and Computer Engineering*, pages 847–850, Piscataway NJ, 2008. IEEE Press.

[36] Advisory Committee on Immunization Practices. Advisory committee on immunization practices policies and procedures. Technical report, Centre for Disease Control and Prevention, 2018.

[37] Sonia Pagliusi, Stephen Jarrett, Benoit Hayman, Ulrike Kreysa, Sai D. Prasad, Martin Reers, Pham Hong Thai, Ke Wu, Youn Tao Zhang, Yeong Ok Baek, Anand Kumar, Anatoly Evtushenko, Suresh Jadhav, Weining Meng, Do Tuan Dat, Weidan Huang, and Samir Desai. Emerging manufacturers engagements in the COVID -19 vaccine research, development and supply. *Vaccine*, 38(34):5418–5423, Jul 2020.

[38] Ana Perisic and Chris Bauch. Social contact networks and disease eradicability under voluntary vaccination. *PLoS computational biology*, 5, 03 2009.

[39] Stanley Plotkin. History of vaccination. *Proceedings of the National Academy of Sciences of the United States of America*, 111(34):12283–12287, Aug 2014.

[40] Robert A. Van Gorder Rahil Sachak-Patwa, Nabil T. Fadai. Understanding viral video dynamics through an epidemic modelling approach. *Physica A*, 502:416–435, February 2018.

[41] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. CreateSpace Independent Publishing Platform, 2016.

[42] James Stewart. *Calculus*. Available 2010 Titles Enhanced Web Assign Series. Cengage Learning, 2007.

[43] Gilbert Syswerda. A study of reproduction in generational and steady state genetic algorithms. In *Foundations of Genetic Algorithms*, pages 94–101. Morgan Kaufmann, 1991.

[44] Punam R Thakare and SS Mathurkar. Modeling of epidemic spread by social interactions. In *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pages 1320–1324. IEEE, 2016.

[45] Meghan Timmins and Daniel Ashlock. Network induction for epidemic profiles with a novel representation. *Biosystems*, 162:205–214, 2017.

[46] John Tregoning, Eric Brown, Hannah Cheeseman, K. E. Flight, S. L. Higham, N.-M. Lemm, B. F. Pierce, D. C. Stirling, Z. Wang, and K. M. Pollock. Vaccines for COVID-19. *Clinical and Experimental Immunology*, 202(2):162–192, Nov 2020.

[47] Anne Tumlinson, William Altman, Jon Glaudemans, Howard Gleckman, and David C. Grabowski. Post-Acute Care Preparedness in a COVID-19 World. *J. Am. Geriatr. Soc.*, 68(6):1150–1154, Jun 2020.

[48] A. M. Turing. Computing machinery and intelligence. *Mind*, 49(236):433–460, Oct 1950.

[49] Douglas West. *Introduction to Graph Theory*. Prentice Hall, Upper Saddle River, NJ 07458, 1996.

[50] Zhaoyang Zhang, Honggang Wang, Chonggang Wang, and Hua Fang. Modeling epidemics spreading on social contact networks. *IEEE transactions on emerging topics in computing*, 3(3):410–419, 2015.