A Pandemic: Avian (bird) Flu

By:

Team: 006

Jesus O. Lopez

Jacob A. Terry

Mathew A. Cardiel

Curtis A. Tade

Sponsoring Teacher: *Albert Simon*

Project Mentor: *Nick Bennett*

Table of Contents:

1.1 Executive Summary

The Avian Influenza, also known as Bird Flu, is a naturally occurring and very contagious virus among birds. There have been hundreds of known cases involving human infection since 1997, caused by contact with infected birds and contaminated objects or through an intermediate host. In humans, the virus causes severe respiratory illness, pneumonia, conjunctivitis, and can even be fatal. An "antigenic shift," in which the avian influenza strain exchanges genes with a human influenza strain to increase its affinity for humans, could result in a possible pandemic, killing millions.

The project was modeled in Java using a network of nodes; the nodes contain a number of factors. Each node is not individual but rather a collective. For example, within a single node data is confirmed by the amount of people within each node, number of carriers with no signs of infection, death rate, and random natural deaths. The nodes simulate a given population, such as a city, community, or neighborhood. Thus the program is flexible for a variety of purposes. The nodes with a higher population density will of course have higher rate of infection than those in a lower population density. The program models the travel behavior of people from one point to another as in everyday life. A person who is infected will be less likely to be traveling in the present condition, whereas a person who is a carrier but not suffering any of the symptoms may travel normally and be infecting other subjects along the way. The models used are assumptions of other similar models that have been collected from the Center of Disease Control. The subjects, meaning the people, are categorized in a non-linear fashion; normal to infected, infected to cured, normal to immune; with some states being skipped altogether.

By running the program we are able to determine the effects on populations, and the specific behavior of the virus in the event of a pandemic outbreak of the Avian Flu in the United States. With data collected from our program's output, epidemiologists can formulate a plan to combat an outbreak of the avian flu. Through future modifications, our program will also be able to be adapted to model the effects and spread of any other disease or pathogen, aiding in the development of a course of action in response to any epidemiological outbreak.

1.2 Introduction

The very possibility of a pandemic should instill concern into people's lives. This caution should not only bring about awareness, but should influence people to take action and attempt to prevent what could be avoided. The Avian Influenza, which was once seen as a minor threat to humans, is now being considered a very likely threat to human life on a large scale. The threat of an Avian Flu Pandemic is all too real, and as with any relevant probability of disaster, the outcome must be foreseen in order to accurately curtail the initial spread. This is the sole purpose for the creation of our project. We desire for people to become aware of what the Avian Flu is and how it will affect them if a pandemic was to occur in the near future. Although only a few hundred cases of human infection have occurred in recent years the adaptation of virus strains (predominantly the H5N1 strain of Influenza A virus) and the possibility of an antigenic shift pose a significant threat. If people were to see a definite proposed outcome of an Avian Flu Pandemic then our project as well as our purpose would be a success.

1.3 Description

This project endeavors to realistically and accurately model, an influenza epidemic within specified bounds or parameters that reflect real-world situations. Additionally, after much research and discussion, a collaborative effort was made by the team to determine realistic parameters to develop the model into one that examines the consequences of an outbreak of Avian Influenza specifically. This model assumes that the virus has experienced an antigenic shift, allowing it to spread by human-to-human contact. Although many critics consider this assumption to be too broad to accurately model an Avian Influenza epidemic, the team felt this was generalization that was necessary to make an attempt to build a model. This model will provide people who work with epidemiological issues with the means to predict the effect of an outbreak. This information is much sought after and extremely important to both professional epidemiologists and the general public. Although the prevention of such an outbreak is the most desirable outcome, the ability to model the effects of it would be a very important aspect of controlling a disastrous situation. This project will allow multiple simulations to be done in a relatively short amount of time. It will help researchers find the best methods of containing the epidemic by letting researchers change the parameters according to their containment methods. Epidemiologists can experiment with different potential solutions through the use of a program such as this one. After running the simulation multiple times, they can evaluate the effect of their ideas by comparing the new results to the previous predicted outcomes. In this way they will be able to determine the most effective methods of limiting the range and effect of the illness.

Of the team members that have done a project of this magnitude in the previous years. One of them had actually done a project very similar to the one know being created. The previous results for the last project were that the team never completed any of the projects due to the time elapse. What the previous did achieve was a creation for the abstract and the interim of the project but without out finding prior research and a working coded program for this simulation.

2.1 Scope

The broad scope of this model is one of its greatest assets. It can model almost any situation simply by changing the parameters to match the real-world situation, i.e. population, population density, the probability of infection, etc. Simply by manipulating these values, entirely new outcomes can be reached, based on the restraints put on the model. The constrain value being the number of Immune, Sick, Total Population, susceptible, Dead, etc. Our program has the capacity to model any infectious illnesses assuming they spread by human-to-human contact only. As stated earlier, in our model of an Avian Influenza epidemic, it is assumed that the virus has mutated to the point where human-to-human contact is the method of spreading the illness. This generalization allows this model to be applied to many other illnesses and this ability greatly enhances the value of the program by making it available to many different people and situations. Other diseases that spread by human-to-human contact can be modeled with this program as long as realistic values are known.

2.2 Materials

This project involves diseases and is, by nature, entirely conceptual in its composition. The only materials needed in the development of this project were research related, including: computers, Internet, paper, books, a great mentor, etc. The programming environment used in this project was Java developed by Sun Microsystems, with great modeling libraries which included Repast agent simulation software for the purpose of making a more user friendly program. We achieved the creativity of this program simulation through the use of our mentor and sponsor.

3.1 Methods

There are a numerous simulation toolkits available now days, requiring various levels of programming experience. One of these, the Repast toolkit, is a set of Java libraries for agent-based modeling (ABM). Repast is quite user-friendly, and many programmers already have experience with Java and C++.

This paper demonstrates the design and implementation of an agent-based, spatially explicit simulation for the study of infectious disease in a human population witch is the Avian Bird Flu. The simulation is readily extensible for multiple applications within this area, and thus can be a good starting point for other researchers. It also demonstrates that the Repast libraries are sufficient to create a realistic, extensible system that is also easy to use for demonstration purposes and researchable topics. The system has enough flexibility and power to model epidemics with an outstanding capability to harness any task. The parameters offer enough detail to be useful for focused numerical experimentation. It is the team's hope that interested researchers will take this code, extend and modify it, and use it for their own research.

3.2 History

Epidemics have been modeled mathematically for over a century. From Louis Pasteur's work on cholera epidemics, an advanced system of partial differential equations was developed to model the change in percentage of a population over time (Wasserman and Faust 1994). From this early model, we get the standard categories used to describe an epidemic. The SIS (Susceptible-Infected- Susceptible) model represents diseases for which there is no immunity; once a person has been infected and recovers, he or she is susceptible to the disease once again. In the SIR (Susceptible-Infected-Removed) model, an individual resides from susceptible to infected to removed (the euphemistic "removed" includes both those with immunity and those that are dead). A more realistic SEIR or SLIR (Susceptible-Exposed/Latent-Infected-Removed) model adds an intermediate step which represents the latent period between exposure and external symptoms and thus can take into account differing degrees of infectiousness which occur during these two stages. Figure 1 shows a visual representation of the various stages used.

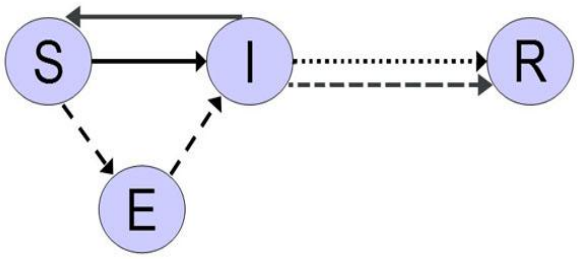


Figure 1. A flowchart of possible states in an epidemic model

The Repast toolkit is a set of libraries in Java, developed to support discrete multi-agent simulations. Repast was developed at _____ a feature of Repast which influenced the decision to use it for this project is its ability to decouple the simulation from real-time visualization, which makes running long batches of simulations much

faster. Repast has many graphical capabilities built in which we have used in the project itself.

This simulation has several parameters that the user can vary. For example, NUM_INFECTED, NUM_REMOVED, and NUM_EXPOSED control the initial number of humans in each of these states. A combination of the NUM_HUMANS parameter with the four display size parameters can be used to set up a specific population density for the simulation. A detailed description of the simulation parameters is shown in the following three tables. Table 1 displays the parameters for the landscape and population of the simulation and their default values.

When the simulation begins, the first step is to apply the parameters to set up its agents and locations. First, the environment is created. By this statement it means to set a user-controlled environment out of inputs from the user. Next, the human agents are created. Agents are chosen to be put in their own city according to the input of the user to fill the required numbers of initially infected, exposed. All other agents are initially set as susceptible to infection. The agents are initially placed randomly occupying. The numbers based on the user or in this case based on the research found for the average people size in a city. When the locations are created the humans are each assigned to a home and a workplace. Sizes of each home and workplace are drawn from an average distribution, which can be modified to mirror various home and workplace size distributions. Homes are placed randomly across the simulation area, while workplaces are confined to the top left quadrant to approximate a city. Now the simulation is ready for the tick to take over. The tick steps the simulation through time, allowing the agents to perform their programmed tasks. The term tick is the meaning of how the user will input the time for an illness to spread overtime. Meaning this could be sec, min, years, decades, millenniums, etc.

3.3 How the program works

The human agents in the simulation lead average humdrum lives: they travel between work and home. Epoch in this simulation can be thought of as 8 AM. At this moment, all human agents begin traveling toward their assigned workplaces. Once they get sufficiently close to their work locations, the human agents each start a timer. When an individual human's timer reaches one third of the day length, it leaves work and travels home. Agents have variable distances to travel, and so arrive at and leave work at different times.

If a human is infected (symptomatic) at epoch, that agent has another class which in a sense flips a coin whether to take a sick day and stay home from work. A random Boolean generator is employed as a coin flip weighted by the simulation's Acceptance parameter.

Influenza (or "the flu") is a commonly known respiratory infection throughout the world. The flu is highly infectious and can be life-threatening to the elderly and those with chronic illness or immunodeficiency. The avian bird flu is contagious in many ways. As present their have been critical cases of the bird flu.

3.4 Mathematical Model

The mathematical model that is presented in this project is a compartmental model. Which is the Expected number of susceptible individuals having significant contact with another individual in the time interval $t < T \leq t + \_t$

Expected number of susceptible individuals having significant contact with infected individuals in the time interval $t < T \leq t + \_t$

Expected number of susceptible individuals who will become infected in the time interval $t \leq T \leq t + \_t$.

Total population

Number of infected at time 0

Number of infected at time t

Number of susceptible at time t

Probability that any given individual has significant contact with any other individual, in a unit time interval2

Probability of infection (i.e. infectiousness), given a single significant contact3

Time step4

We can now express the number infected at any time as the sum of those infected at the end of the previous time step, and the new infections1. Since we are using an SI model, where all members of a population are assumed to be either infected or susceptible, we can estimate the number susceptible at any given time as the difference between the total population and the estimate of the number infected at that time. Alternatively, we can estimate the number susceptible in the same fashion as the number infected, by virtue of the fact that new infections flow out of the S compartment at the same time that they flow into the, I compartment. We can also say that the difference between the number infected at time $t + \_t$ and the number infected at time t is the flow into the I compartment, which is itself a function of the time difference ($\_t$). This general form, where one difference is expressed as a function of another, gives us the term "difference equation". We now have formulas for estimating the number infected and the number susceptible at the end of each time step, in terms of the same quantities at the end of the previous time step.

● We assume that we know the initial number infected (I0), the total population (P), and thus the initial number susceptible (S0 = P – I0).
● We can now estimate the number infected and the number susceptible at any given time (It and St), by starting at time 0, and moving forward repeatedly by the time step, calculating the number infected and number susceptible at each step, until we reach the moment of interest.

As mentioned previously, the selection of a step size is a non-trivial issue. For example, even though the chart on page 8 is pretty smooth, we can see (if we look closely) that it is not actually a straight line, nor is it the smooth curve that page 9 might lead us to believe. Making the step size smaller (computing a correspondingly larger number of steps) would make it even smoother; what might not be as obvious is that the resulting curve could change in more substantive ways with different step sizes.
● In fact, when the flow rate between two compartments is not constant, but depends on the level in one or the other compartment, different step sizes will give different results – sometimes subtly so, but sometimes dramatically so.
● Above all, we need to select a step size which is appropriate for the real world problem we are modeling. For example, if we are modeling galactic formation, a step size of a second, an hour, or even a day would be too small: we would spend all our time calculating almost insignificant changes. On the other hand, if we are modeling quantum particle collisions, a step size of a second is many times too large: we are modeling changes that take place in tiny fractions of a second, and our step size must be set accordingly.

For an epidemic model, a step size of a day (or some significant fraction of a day) is generally appropriate.

● When we choose a step size which is equal to the unit time – as in our example, where we are using a unit time of 1 day, and our step size is 1 day – it is fairly common to drop the _t symbol from the equations of flow, and from the resulting difference equations.1

In estimating the number infected, we quickly start getting numbers which aren't integers. In epidemic modeling, we can argue that it doesn't make sense to consider a fraction of an individual to be infected: either an individual is infected, or it isn't. So we often round the results of each step's calculations to the nearest integers, and use these rounded values to compute the next step. (Note that this is not the same as doing the computations for all of the steps – out to the time horizon of interest – without rounding, and only rounding the results at the end.)

● On the other hand, there are some cases in epidemic modeling where we definitely wouldn't round the results. For example, if we were mostly interested in the proportions of the population in the S and I compartments at each step, rather than the actual numbers, then we could represent the total population as 1.0, and the initial infected as some fractional value (in our example, it would be 0.001), and not perform any rounding. Also, if we were to apply calculus to our model, to analyze what happens as we make the step size ever smaller (approaching a limit of 0), we would need to avoid the stair-step effect produced by rounding.

Even when we decide to round our results at each step, the appropriate direction for the rounding is not always clear. Should we round to the nearest integer? Always round down? Always round up? The answer is not the same in every case, and it depends greatly on the type of problem being modeled.

● Finally, we must be aware that in many cases, the results produced with rounding are qualitatively different than those produced without rounding.

In particular, our flow equations and step size might be such that rounding will result in no change at all in compartment levels from step to step; this might be appropriate, or it might be an indicator that our step size is too small (or maybe even that we shouldn't be rounding).

Appendix A

Sources:

http://www.historylearningsite.co.uk/louis_pasteur.htm - Louis Pasteur

http://217.222.182.72/html/elexcourse/htm/lessons/L4_5.htm - Wasserman and Faust

Appendix B: Information

Louis Pasteur

Louis Pasteur was born in 1822 in Dole, France. Louis Pasteur's name is forever cemented in the history of medicine. He, along with Alexander Fleming, Edward Jenner, Robert Koch and Joseph Lister, is of great importance when studying medical history. Pasteur's discovery – that of germs – may seem reasonably tame by the standards of 2002, but his discovery was to transform medicine and see his name forever immortalized on a day-to-day basis in pasteurized milk – named in his honor.

Pasteur is important for three reasons:

Pasteur showed that airborne microbes were the cause of disease. Pasteur built on the work of Edward Jenner and helped to develop more vaccines Pasteur's career showed how conservative the medical establishment was at the time.

As a young man, Pasteur studied at the Ecôle Normale in Paris. In 1843, he became a research chemist. He developed such a reputation that in 1854 aged just 32, he became Dean of the Faculty of Science at the University of Lille. At this time, Lille was the centre of alcohol manufacture in France. IN 1856, Pasteur received a visit from a man called Bigo who worked at a factory that made alcohol from sugar beet. Bigo's problem was that many of his vats of fermented beer were turning sour and, as a result, the beer had gone off and had to be thrown away. From a business point of view, this was a disaster. Bigo asked Pasteur to find out why this was happening.

After using a microscope to analyze samples from the vats, Pasteur found thousands of tiny micro-organisms. He became convinced that they were responsible for the beer going sour. Pasteur believed that they caused the putrefaction of the beer – not that they were the result of the putrefaction.

Pasteur continued his work on this theme by studying other liquids such as milk, wine and vinegar. In 1857, he was appointed Director of Scientific Studies at the Ecôle Normale in Paris. Between 1857 and 1859, Pasteur became convinced that the liquids he had studied were being contaminated with microbes that floated in the air. The medical establishment ridiculed him:

> **"I am afraid that the experiments you quote, M. Pasteur, will turn against you. The world into which you wish to take us is really too fantastic."**
>
> **La Presse, 1860**

Pasteur was vilified in public but rather than give up, he determined to fight for what he believed in. Pasteur started to devise tests to prove that he was right. He was able to prove that:

Air contained living organisms that these microbes can produce putrefaction that these microbes could be killed by the heating of the liquid they were in That these microbes were not uniformly distributed in the air.

In April 1864, Pasteur explained his beliefs in front of a gathering of famous scientists at the University of Paris. He proved his case beyond doubt – even if some of those present refused to believe him including Dr. Charlton Bastian who maintained his belief that putrefaction came from within and not from invading micro-organisms.

Up to 1865, Pasteur's work only involved beer, wine and milk. In 1865, he was asked to investigate his first disease called pébrine that affected the silk worm industry. Within a year, Pasteur had established that the disease was caused by a living organism and he now became convinced that microbes could also affect humans as well as beer and silk worms. In this sense, Pasteur believed that microbes could spread diseases among humans. Three of Pasteur's daughters had died between 1859 and 1865; two from typhoid and one from a brain tumor.

In 1865, a cholera epidemic hit Marseilles. Pasteur carried out a number of experiments in a hospital in the hope of finding the germ that caused this feared disease. He was not successful.

In 1868, Pasteur suffered from a brain hemorrhage that affected the left side of his body. This affected his ability to work but the work that he had done up to 1868, had inspired a number of younger scientists.

Pasteur developed his work by finding out ways humans could be prevented from getting a disease. He was inspired by his own desire to develop his knowledge but also by patriotism. Robert Koch was getting a great deal of attention throughout Europe for his discoveries and the French versus German rivalry that occurred provided a great spur to medical advances. In 1881, Pasteur met Koch at a meeting in London when the German was giving a lecture on what he had discovered up to that date. All Pasteur said to Koch after the lecture was "That is great progress".

Koch had gathered around him a team of skilled research scientists. Pasteur frequently worked by himself. He realized that this was not the way to proceed and he also gathered around him a team of research scientists. Pasteur had always lacked detailed medical knowledge. Because of this he introduced into his team two brilliant young doctors, Emile Roux and Charles Chamber land. The first disease this team worked on was chicken cholera – a disease that affected many poultry farmers.

Pasteur knew about the work done by Edward Jenner regarding smallpox. Pasteur reasoned that if a vaccine could be found for smallpox, then a vaccine could be found for all diseases. Pasteur did not know how Jenner's vaccination worked so he had to proceed searching for a chicken cholera vaccine using a process of trial and error.

In the summer of 1880, he found a vaccine by chance. Chamber land had inoculated some chickens with chicken cholera germs from an old culture that had been around for some time. The chickens did not die. Pasteur asked Chamber land to repeat what he had done but with a fresh culture of chicken cholera germs. Pasteur reasoned that a new culture would provide more potent germs.

Two groups of chickens were inoculated; one that had been given the old culture and one group that had not. Those chickens that had been given the old culture survived those that had not died. The chickens that had been inoculated with the old culture had

become immune to chicken cholera. Pasteur believed that their bodies had used the weaker strain of germ to form a defense against the more powerful germs in the fresher culture.

In April 1881, Pasteur announced that his team had found a way to weaken anthrax germs and so could produce a vaccine against it. Despite his fame, there were still those in the medical world who mocked Pasteur.

> **"Will you have some microbe? There is some everywhere. Microbiolatry is the fashion, it reigns undisputed; it is a doctrine which must not even be discussed, especially when its Pontiff, the learned Monsieur Pasteur, has pronounced the sacramental words, "I have spoken". The microbe alone is and shall be the characteristic of a disease; that is understood and settled; the Microbe alone is true, and Pasteur is its prophet**."
>
> Rossignol, written in 1881.

Rossignol was the editor of "The Veterinary Press" and in 1882 he challenged Pasteur to a public test of his anthrax vaccine. The tests were held in May 1882. Sixty sheep used in the test. Pasteur kept ten as they were and divided the other fifty into two groups of twenty-five. One group was inoculated with his vaccine while twenty-five were not. All fifty were then injected with the anthrax virus. Those that were not inoculated died within two days. The inoculated group suffered no ill-effects and were described as being "sound, and (they) frolicked and gave signs of perfect health". They proved that Pasteur was not exaggerating the powers of his vaccine. "The Times" in Great Britain called Pasteur "one of the scientific glories of France".

Pasteur and his team turned next to the disease of rabies. Most human victims of rabies died a painful death and the disease appeared to be getting more and more common in France. Though the team could not identify the germ, they did find that the rabies germ attacked the nervous system only after it had made its way to the brain. The team traced the germ to the brain and spinal cord of infected animals and by using dried spinal cords; they produced a vaccine for rabies. The vaccine was first tried out on animals.

Pasteur injected ‘clean’ animals with the rabies germ found in spinal cord that was fourteen days old. At this age, the germ was relatively weak and unlikely to threaten the life of the animals. He then used spinal cords that were thirteen days old, twelve days etc. on the animals until they were injected with the most virulent germ found in infected spinal cord that was fresh. All survived this. But Pasteur faced a serious problem. What worked on animals might not work on humans.

In 1885, a young boy, Joseph Meister, had been bitten by a rabid dog, and was brought to Pasteur. The boy almost certainly would have died an agonizing death if nothing was done so Pasteur took the risk on using his untested vaccine.

"The death of this child appearing to be inevitable, I decided, not without lively and sore anxiety, as may well be believed, to try upon Joseph Meister, the method which I had found constantly successful with dogs. Consequently, sixty hours after the bites and in the presence of Drs Vulpian and Grancher, young Meister was inoculated under a fold of skin with half a syringe of the spinal cord of a rabbit, which had died of rabies. It had been preserved (for) fifteen days in a flask of dry air. In the following days, fresh inoculations were made. I thus made thirteen inoculations. On the last days, I inoculated Joseph Meister with the most virulent virus of rabies."

Pastuer

The boy survived and Pasteur knew that he had found a vaccine for rabies. Three months later, when he examined Meister again, Pasteur reported that the boy was in good health.

Ironically, though Pasteur and his team knew that the vaccine worked, no one then in the world of science knew how it worked!

Wasserman and Faust

Wasserman and Faust (1994), which makes a distinction between the structural and positioning properties of a network, which we will be describing in this paragraph, and the roles and positions of the actors that comprise it, which we will be discussing in the following paragraphs. Amongst the main structural and positioning properties of a network, we will consider some dimensions that are particularly suitable for the description of virtual communities: centrality and prestige, as well as the cohesion of the sub-groups (clique, n-clique, etc.).

The identification of the importance taken on by the various actors within a social network has always been one of the main aims of social network analysis. The importance of the actors within a relationship network can be shown by their centrality and prestige. These indices "seek to describe and measure the properties of the 'actor's position' within a social network" and the measurements regarding the individual actors can then be combined in order to measure the centrality or prestige of the network as a whole (Wasserman and Faust, 1994).

1. Centrality and centralization

Centrality identifies the "most central", "most important" or "most significant" actors in a social network and has been one of the main indices to be considered by SNA scholars. As we will see, centrality is not defined by a single index, but rather by several indices in correspondence to structural aspects of the interactions that the researcher intends to focus on. The formulae and calculation of some indices are particularly complex and their definition and perfection is certainly primarily due to L.C. Freeman. The aim of this treatment is not the detailed description of the mathematical principles underlying the various indices, but rather the understanding of their use in order to analyse the structure of the interactions within virtual communities. The following pages

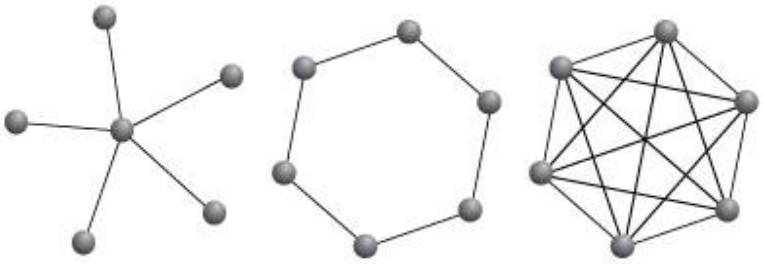
therefore focus on the question of **what can be described using centrality indices** , while readers who wish to examine the mathematical principles further are referred to the articles by L.C. Freeman and the volume by Wasserman and Faust (1994).

Firstly, it is necessary to make a distinction between **point centrality indices** and **centralization indices**[1] or graph centrality indices (Wasserman and Faust, 1994; Scott, 1997). The point (node or actor) **centrality index** expresses the strategic importance of a certain point for the overall graph structure, or rather its importance with respect to the entire relationship network. As it is a specific measurement of individual nodes, the centrality index makes it possible to check whether there are any differences between the various nodes in relation to their significance for the structure of the relationship network and, if necessary, to identify the most central nodes and the most peripheral nodes. The value of this index varies from a minimum of Ø (extremely peripheral nodes or points) to a maximum of 1 (extremely central nodes or points). Unlike the centrality indices, the **centralization indices** regard the entire structure of a graph and describe how it is centralized around its most central points. Centralization may also be considered a measurement of the variability and dispersion of the centrality indices of the individual nodes (Wasserman and Faust, 1994).

In fact, "The general procedure for measuring the centralization of a graph involves looking at the differences between the centrality of the most central point and that of all the other points. **Centralization is therefore the relationship between the real sum of the differences and their potential maximum sum"** (Scott, 1997).

Like centrality indices, centralization indices also vary from a minimum of Ø to a maximum of 1, but can also be expressed in percentages from Ø to 100%. The above shows how, with the increase of the centralization index, the differences between the centrality indices of the individual points also increase and there is a greater probability that an individual point or a few points will be found in a very central position, while with the decrease of the centralization index, the differences between the centrality indices are reduced and it is likely that there are no particularly central nodes with respect

to others. The following examples will provide a clearer picture of this relationship between point centrality indices and the graph centralization index.



Star graph Circle graph Complete graph

The **star graph** is characterized by a single central point and a number of peripheral points. The central point will have a top value centrality index, inasmuch as it is the only one to have links with all the other nodes, while the latter have just one link each with the central point. Since the difference between the central point centrality index and the peripheral point centrality indices is huge, the centralization index will be of the top value, or 1.

The **circle graph**, on the other hand, features points that all have just two links and no point in a central position. The individual point centrality indices will all be the same, while the centralization index will have a minimum value, or rather Ø.

Finally, the **complete graph** is in the same situation as the circle graph (all the points have the same centrality index, although in this case every point is connected to all the other points in the graph. Therefore, in the complete graph too, the centralization index will be Ø.

The centralization index therefore tells us that the star graph is totally centralized around its most central point, while the circle graph and the complete graph are not centralized around any one point. Like the density index, the centralization indices describe some aspects of the "overall compactness of a graph" (Scott, 1997, p. 131).

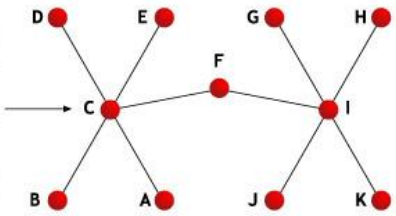
“The density describes the general level of cohesion in a graph; its centralization describes the extent to which this cohesion is organized around particular focal points. The centralization and density are therefore important, complementary measurements” (Scott, 1997).

The centrality indices that we consider in this treatment[2] are the **degree centrality,** the **closeness centrality** and the **betweens centrality.**

As far as regards the centralization indices, we will consider the **degree centralization index** , the **closeness centralization index** and the **betweens centralization index.** Let's start by considering dichotomous (valueless) and symmetrical (unoriented) relationships. We will then go on to consider the applicability of centrality and centralization indices to oriented, valued graphs.

In order to describe the specific characteristics of the individual indices, we will take a hypothetical group of 11 researchers, of different nationalities, who communicate and collaborate by e-mail on an international research project. The following figure features the adjacency matrix of the dichotomized (valueless) and symmetrical (unoriented) relationship data and the relative graph obtained using **NetMiner** software[3]

	A	B	C	D	E	F	G	H	I	J	K
A		0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
B	0.0		1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
C	1.0	1.0		1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
D	0.0	0.0	1.0		0.0	0.0	0.0	0.0	0.0	0.0	0.0
E	0.0	0.0	1.0	0.0		0.0	0.0	0.0	0.0	0.0	0.0
F	0.0	0.0	1.0	0.0	0.0		0.0	0.0	1.0	0.0	0.0
G	0.0	0.0	0.0	0.0	0.0	0.0		0.0	1.0	0.0	0.0
H	0.0	0.0	0.0	0.0	0.0	0.0	0.0		1.0	0.0	0.0
I	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0		1.0	1.0
J	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0		0.0
K	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	



By observing the graph, we can see that all the researchers are involved in the interactions (100% inclusivity), although many of them (A, B, D, E, G, H, J, K) only have contact with one other researcher. Because of this, the density is rather low

$$\frac{1}{n\,(n\text{-}1)\,/2} = \frac{10}{11\,(11\text{-}1)\,/\,2} = 0{,}182$$

and the network is equally vulnerable (line-connectivity = 1): in fact, it is sufficient to eliminate just one of the two links that connect researcher F to researchers C and I in order to separate the network.

Apendix C: Source Code

```
package org.supercomputingchallenge.sti.epidemic;

import org.supercomputingchallenge.sti.systemdynamics.Model;
import org.supercomputingchallenge.sti.systemdynamics.Stock;

public class RunBasicSIModel {

   private static final String SUSCEPTIBLE_NAME = "Susceptible";
   private static final String INFECTED_NAME = "Infected";
   private static final double CONTACT_PROBABILITY = 0.2d;
   private static final double INFECTION_PROBABILITY = 0.25d;
   private static final double TOTAL_POPULATION = 100000d;
   private static final double INITIAL_INFECTED = 100d;
   private static final double DELTA_TIME = 1d;
   private static final double MAX_TIME = 500d;
   private static final double EPSILON = 0.5d;

   public static void main(String[] args) {
      Model model = new Model(DELTA_TIME);
      Stock susceptible =
         new Stock(SUSCEPTIBLE_NAME, TOTAL_POPULATION -
INITIAL_INFECTED);
      Stock infected = new Stock(INFECTED_NAME, INITIAL_INFECTED);
      model.addStock(susceptible);
      model.addStock(infected);
      model.addFlow(new Infection(susceptible, infected,
                  CONTACT_PROBABILITY,
INFECTION_PROBABILITY));
      Monitor reporter = new Monitor(System.out, SUSCEPTIBLE_NAME,
```

```
                    INFECTED_NAME, MAX_TIME, EPSILON);
            model.run(reporter);
        }

    }

    package org.supercomputingchallenge.sti.epidemic;

    import uchicago.src.sim.engine.SimInit;

    public class RunRepastSIModel {

        public static void main(String[] args) {
            SimInit sim = new SimInit();
            sim.loadModel(new RepastSIModel(), null, false);
        }

    }
package org.supercomputingchallenge.sti.epidemic;

import uchicago.src.sim.engine.SimInit;

public class RunRepastSIModel {

   public static void main(String[] args) {
      SimInit sim = new SimInit();
      sim.loadModel(new RepastSIModel(), null, false);
   }

}
```

```
package org.supercomputingchallenge.sti.epidemic;

import java.util.Map;
import org.supercomputingchallenge.sti.systemdynamics.Flow;
import org.supercomputingchallenge.sti.systemdynamics.Stock;

public class Recovery extends Flow {

   private double recoveryRate;

   public Recovery(final Stock infected, final Stock recovered,
         double recoveryRate) {
      super(infected, recovered);
      this.recoveryRate = recoveryRate;
   }

   public double getRate(
         final Map<String, Stock> stocks, final double currentTime) {
      double numInfected = getSource().getLevel();
      return (numInfected * getRecoveryRate());
   }

   public double getRecoveryRate() {
      return recoveryRate;
   }

   public void setRecoveryRate(final double recoveryRate) {
      this.recoveryRate = recoveryRate;
   }

}
```

```
package org.supercomputingchallenge.sti.epidemic;

import java.io.PrintStream;
import java.util.Map;
import org.supercomputingchallenge.sti.systemdynamics.Model;
import org.supercomputingchallenge.sti.systemdynamics.Stock;

public class Monitor
      implements org.supercomputingchallenge.sti.systemdynamics.Monitor {

   protected static final String SNAPSHOT_HEADER =
         "\"Time\",\"Susceptible\",\"Infected\"\r\n";
   protected static final String SNAPSHOT_PATTERN = "%f,%f,%f\r\n";

   private PrintStream output;
   private String susceptibleName;
   private String infectedName;
   private double maxTime;
   private double epsilon;

   public Monitor(final PrintStream output,
         final String susceptibleName, final String infectedName,
         final double maxTime, final double epsilon) {
      this.output = output;
      this.susceptibleName = susceptibleName;
      this.infectedName = infectedName;
      this.maxTime = maxTime;
      this.epsilon = epsilon;
      output.print(SNAPSHOT_HEADER);
   }
```

```
    public boolean report(final Model model) {
        Map<String, Stock> stocks = model.getStocks();
        double time = model.getCurrentTime();
        double susceptible = stocks.get(susceptibleName).getLevel();
        double infected = stocks.get(infectedName).getLevel();
        output.printf(SNAPSHOT_PATTERN, time, susceptible, infected);
        return ((time < maxTime) && (susceptible >= epsilon));
    }

}

package org.supercomputingchallenge.sti.epidemic;

import java.io.PrintStream;
import java.util.Map;
import org.supercomputingchallenge.sti.systemdynamics.Model;
import org.supercomputingchallenge.sti.systemdynamics.Stock;

public class Monitor
        implements org.supercomputingchallenge.sti.systemdynamics.Monitor {

    protected static final String SNAPSHOT_HEADER =
            "\"Time\",\"Susceptible\",\"Infected\"\r\n";
    protected static final String SNAPSHOT_PATTERN = "%f,%f,%f\r\n";

    private PrintStream output;
    private String susceptibleName;
    private String infectedName;
    private double maxTime;
    private double epsilon;
```

```
    public Monitor(final PrintStream output,
            final String susceptibleName, final String infectedName,
            final double maxTime, final double epsilon) {
        this.output = output;
        this.susceptibleName = susceptibleName;
        this.infectedName = infectedName;
        this.maxTime = maxTime;
        this.epsilon = epsilon;
        output.print(SNAPSHOT_HEADER);
    }

    public boolean report(final Model model) {
        Map<String, Stock> stocks = model.getStocks();
        double time = model.getCurrentTime();
        double susceptible = stocks.get(susceptibleName).getLevel();
        double infected = stocks.get(infectedName).getLevel();
        output.printf(SNAPSHOT_PATTERN, time, susceptible, infected);
        return ((time < maxTime) && (susceptible >= epsilon));
    }

}

package org.supercomputingchallenge.sti.epidemic;

import java.util.Map;
import org.supercomputingchallenge.sti.systemdynamics.Flow;
import org.supercomputingchallenge.sti.systemdynamics.Stock;

public class Infection extends Flow {

    private double contactProbability;
```

```
private double infectionProbability;

public Infection(final Stock susceptible, final Stock infected,
        final double contactProbability,
        final double infectionProbability) {
    super(susceptible, infected);
    this.contactProbability = contactProbability;
    this.infectionProbability = infectionProbability;
}

public double getRate(
        final Map<String, Stock> stocks, final double currentTime) {
    double numSusceptible = getSource().getLevel();
    double numInfected = getSink().getLevel();
    double numTotal = numSusceptible + numInfected;
    double proportionSusceptible = numSusceptible / numTotal;
    double proportionInfected = numInfected / numTotal;
    return (proportionSusceptible * proportionInfected
            * contactProbability * infectionProbability * numTotal);
}

public double getContactProbability() {
    return contactProbability;
}

public void setContactProbability(final double contactProbability) {
    this.contactProbability = contactProbability;
}

public double getInfectionProbability() {
    return infectionProbability;
```

```
    }

    public void setInfectionProbability(final double infectionProbability) {
        this.infectionProbability = infectionProbability;
    }

}
```