2011

Using Social Networks for Modeling and Optimization in a Healthcare Setting

Donald Ephraim Curtis
University of Iowa

USING SOCIAL NETWORKS FOR MODELING AND OPTIMIZATION IN A HEALTHCARE SETTING

by

Donald Ephraim Curtis

An Abstract

Of a thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Computer Science
in the Graduate College of
The University of Iowa

July 2011

Thesis Supervisor: Associate Professor Sriram Pemmaraju

ABSTRACT

Social networks encode important information about the relationships between individuals. The structure of social networks has important implications for how ideas, information, and even diseases spread within a population. Data on online social networks is becoming increasingly available, but fine-grained data from which physical proximity networks can be inferred is still a largely elusive goal. We address this problem by using nearly 20 million anonymized login records from University of Iowa Hospitals and Clinics to construct healthcare worker (HCW) contact networks. These networks serve as proxies for potentially disease-spreading contact patterns among HCWs. We show that these networks exhibit properties similar to social networks arising in other contexts (e.g., scientific collaboration, friendship, etc.) such as the "Six Degrees of Kevin Bacon" (i.e., small-world) phenomenon. In order to develop a theoretic framework for analyzing these HCW contact networks we consider a number of random graph models and show that models which only pay attention to local structure may not adequately model disease spread. We then consider the best known approximation algorithms for a number of optimization problems that model the problem of determining an optimal set of HCWs to vaccinate in order to minimize the spread of disease. Our results show that, in general, the quality of solutions produced by these approximations is highly dependent on the dynamics of disease spread. However, experiments show that simple policies, like vaccinating the most well-connected or most mobile individuals, perform much better than a random vaccination policy. And finally we consider the problem of finding a set of individuals to act as indicators for important healthcare related events on a social network for infectious disease experts. We model this problem as a generalization of the *budgeted maximum coverage* problem studied previously and show that in fact our problem is much more difficult to solve in general. But by exposing a property of this network, we provide analysis showing that a simple greedy approach for picking indicators provides a near-optimal (constant-factor) approximation.

Abstract Approved: ______________________________
Thesis Supervisor

____
Title and Department

____
Date

USING SOCIAL NETWORKS FOR MODELING AND OPTIMIZATION IN A HEALTHCARE SETTING

by

Donald Ephraim Curtis

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Computer Science
in the Graduate College of
The University of Iowa

July 2011

Thesis Supervisor: Associate Professor Sriram Pemmaraju

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph.D. thesis of

Donald Ephraim Curtis

has been approved by the Examining Committee for the thesis requirement for the Doctor of Philosophy degree in Computer Science at the July 2011 graduation.

Thesis Committee:

Sriram Pemmaraju, Thesis Supervisor

Alberto Segre

Philip Polgreen

Ted Herman

Kasturi Varadarajan

To Tara

TABLE OF CONTENTS

LIST OF TABLES v

LIST OF FIGURES vii

CHAPTER

1 INTRODUCTION 1

1.1 Hospital Acquired Infections 3
1.2 Contact Network Epidemiology 4
1.3 Random Graph Models for Contact Networks 7
1.4 Vaccination Policies 9
1.5 Disease Surveillance 11

2 GENERATING HCW CONTACT NETWORKS 15

2.1 Constructing HCW Contact Networks 15
2.1.1 The Hospital Graph 17
2.1.2 Extracting contacts 20
2.1.3 HCW contact networks: discussion 22
2.2 Computers-People Graph 24
2.3 Analysis of HCW Contact Networks 25
2.3.1 Degree Distributions 26
2.3.2 Community Structure 29
2.3.3 Small World Property 31
2.3.4 Diversity within Groups of Healthcare Workers 32
2.3.5 Discussion 34

3 RANDOM GRAPH MODELS OF CONTACT NETWORKS 36

3.1 Chapter Overview 40
3.2 Graph Metrics 41
3.3 Generating Random Graphs 45
3.3.1 Comparison of Graph Properties 49
3.4 Comparing the Spread of Disease 54
3.5 Clustering in Networks 67
3.5.1 Construction-Based Models 71
3.5.2 Edge Swapping 79
3.6 Spatial-Clustering Model 90
3.7 Discussion 97

4 VACCINATION POLICIES 99

4.1 Vaccination Problem 99
4.2 Budgeted Vaccination Problem 100
4.2.1 Sum-of-Squares Partition Problem 103
4.2.2 Budgeted Vertex Cover 106
4.2.3 Min-Max Degree 110
4.3 Restricted Disease Problem 113
4.3.1 Partial Vertex Cover 114
4.3.2 Restricted Max Degree 118
4.3.3 Discussion 122
4.4 Experimental Analysis of Simple Vaccination Policies 124
4.4.1 Improving on Simple Heuristics 140

5 BUDGETED MAXIMUM COVERAGE 148

5.0.2 Results 154
5.1 The Reward-Cost Model 155
5.1.1 Choosing Important Threads 155
5.1.2 Criteria for Participation 157
5.2 A Greedy Algorithm for BMOC 157
5.2.1 BMOC is Difficult to Approximate 159
5.2.2 The Overlap Condition 163
5.3 Experiments on BMOC 168
5.3.1 Greedy Performance 169
5.3.2 Analysis of Selected Users 170
5.3.3 Analysis of Selected Threads 173
5.3.4 An Upper-Bound to **Greedy** 175

6 CONCLUSIONS AND FUTURE WORK 180

6.1 Generating HCW Contact Networks 180
6.2 Random Graph Models of Contact Networks 181
6.3 Vaccination Policies 182
6.4 Budgeted Maximum Coverage 183

REFERENCES 184

LIST OF TABLES

Table

2.1 The largest ten groups of HCWs partitioned by distinct (department, job title) pairs. 33

2.2 The number of groups for sizes from 1 to 10. 33

3.1 Graph statistics for the $sparse_{50}$ graph and graphs generated using the Erdös-Rényi (ER), Barabási-Albert (BA), Chung-Lu (CL), Configuration (CONF), and Configuration with Assortativity (CA) models. 49

3.2 Graph statistics for the $moderate_{50}$ graph and graphs generated using the Erdös-Rényi (ER), Barabási-Albert (BA), Chung-Lu (CL), Configuration (CONF), and Configuration with Assortativity (CA) models. 50

3.3 Numerical comparison of the curves in Figure 3.3 measured by the sum of the squared difference in the number of individuals infected each day. . . 56

3.4 Statistics for the plots in Figures 3.3 and 3.4 showing mean and median number of individuals infected, the mean positive increase in the curve and day with the most number of infected individuals. 59

3.5 Statistics for the plots in Figure 3.5 showing mean and median number of individuals infected, the mean positive increase in the curve and day with the most number of infected individuals. 60

3.6 Numerical comparison of the curves in Figure 3.5 measured by the sum of the squared distances each day. 60

3.7 Mean number of infected individuals for the disease curves in Figure 3.6. 63

3.8 Mean number of infected individuals for the disease curves in Figure 3.7. 65

3.9 Table comparing the number of triangles, actual edges, and minimum required triangle edges if no two triangles share an edge for two real-world networks and our HCW contact networks. 73

3.10 Results from experimental runs of the **DEG** algorithm. 76

3.11 Clustering statistics for graphs generated based on the BKM model. . . 84

3.12 Graph statistics for the $moderate_{50}$ graph and graphs generated Spatial-Clustering (SC) with $\gamma = 3$, CONF, and Configuration with Assortativity models. 91

3.13 Statistics for the plot in Figure 3.22 showing mean and median number of individuals infected and the mean number of new infections each day. . 93

3.14 Numerical comparison of the curves in Figure 3.22 measured by the sum of the squared distances each day. 94

3.15 Statistics for the plot in Figure 3.23 showing mean and median number of individuals infected and the mean number of new infections each day. . 95

3.16 Numerical comparison of the curves in Figure 3.23 measured by the sum of the squared distances each day. 96

4.1 Parameters used in testing the mincut-degree-based policy. 143

4.2 Statistics for the graph resulting from removing edges returned by the mincut process in Algorithm 4.1. 144

5.1 Results from analyzing the overlap condition for user subsets $U' \subseteq U$ of *pairs* ($|U'| = 2$) and *triples* ($|U'| = 3$) in the user-thread graph with $thresh = 80$. 169

5.2 The cost of solutions that achieve 50%, 75%, and 100% coverage of important threads (corresponding to points from the plots shown in Figure 5.4). 171

5.3 Statistics for sample runs of **Greedy** algorithm. 172

5.4 EIN traffic statistics for the year 2007 for full coverage at $thresh = 80$. . 175

LIST OF FIGURES

Figure

2.1 The first five of approximately 19.8 million EMR login records. The `UserIDs` are all de-identified, although each de-identified user has an associated `position & dept` field. The `Device` field provides computer IDs with associated `Location` information. The `Location` field specifies rooms in the UIHC (e.g., `RCP 1100` is room number 1100 in the Roy Carver Pavilion of the hospital). Note that some of the records are missing the `Device` field, rendering them unusable for contact graph construction. needed for contact network construction, still leaving about 11.7 million usable records. 16

2.2 Table showing the size and other aggregate characteristics of the EMR login data. 17

2.3 A CAD drawing fragment for the basement (floor 0) of the hospital, showing how it was marked up by hand in order to break up corridors into segments that were approximately room-sized. 19

2.4 A small portion of the hospital graph, corresponding to the second floor of the UIHC. The inset makes clear how each room or corridor segment is represented by a vertex, connected by edges to adjacent rooms or corridor segments. This particular image was produced by superimposing the graph onto a CAD drawing of the floor plan. 19

2.5 This picture shows the entire hospital graph superimposed on a 3-dimensional architectural drawing of the hospital. The vertices are colored according their building designator. 20

2.6 Basic characteristics of the hospital graph. This graph has an average degree of 1.236, which is consistent with our observation that most rooms have degree one or two because they connect only to a corridor or to a corridor and a bathroom. 20

2.7 The different parameters and their possible values that we use for generating healthcare worker contact networks. With 5 values for d, 5 for t, and 90 for T, all independently chosen, we obtain over 2,250 different healthcare contact networks. 22

2.8 Small section of HCW contact networks generated based on EMR login data using different definitions of a contact. (a) Contact graph generated with $d = 1, t = 0$. (b) Contact graph generated with $d = 3, t = 15$. (c) Contact graph generated with $d = 5, t = 30$. 22

2.9 These statistics show that the computers-people graph is relatively sparse However, both person-degrees and computer-degrees show a large standard deviation raising the possibility of a few heavily used computers and a few highly mobile healthcare workers. 24

2.10 Small portion of a computers-people graph for a 4-week period. Computers are marked by black dots and are shown in their actual location on the 4th floor of the UIHC. Healthcare workers are shown as white dots and the edges connect healthcare workers to the computers they have logged into during the 4-week time window. 25

2.11 Basic structural features of a $sparse_1$ ($d = 1$, $t = 0$), $moderate_1$ ($d = 3$, $t = 15$), and $dense_1$ ($d = 5$, $t = 30$) instance of the HCW contact network are shown here. Note that the dense graph is only dense relative to the sparse graph; the average degree of even the dense graph is less than 1% of the graph size. For all three graphs we use time window $T = 001$, i.e., a 4-week time window starting at the second week of our EMR login data. For comparison, the corresponding statistics for Erdös-Renyi random graphs with same size (n) and same mean degree ($\langle k \rangle$) are also provided. 27

2.12 Degree distribution of the $moderate_1$ HCW contact network and max likelihood power-law (dashed line) and log-normal (light curve) fits. (a) log-log plot; each point $(d, p(d))$ represents the fraction $p(d)$ of healthcare workers with degree d in the moderate HCW contact network. (b) cumulative form; each point $(d, c(d))$ represents the fraction $c(d)$ of healthcare workers with degree at most d) is shown here. 28

2.13 Modularity values for partitions of $sparse_1$, $moderate_1$, and $dense_1$ graphs obtained via different methods. The modularity of the partition induced by "job class" falls well below the 0.3 threshold for a strong community structure, mentioned by Newman [80]. The last row shows modularity values for partitions obtained by using a greedy clustering algorithm, called `maxQ`, due to Clauset et al. [29] 30

2.14 Cumulative density plots for the (a) degree distribution and (b) distance distribution for each of the largest 10 groups partitioned by (department, job title) pairs. The plots show, for a particular x-value, the fraction of healthcare workers whose degree (respectively, distance traveled) is at most the x-value. Thickness of the line indicates the size of the group. . 34

3.1 Degree distributions for the $sparse_{50}$ HCW contact network and corresponding graphs generated by the Erdös-Rényi (ER), Chung-Lu (CL), and Barabási-Albert (BA) models. x-axis denotes the degree. y-axis denotes the fraction of nodes. Plot is truncated; maximum values can be found in Table 3.1. 52

3.2 Degree distributions for the $moderate_{50}$ HCW contact network and corresponding graphs generated by the Erdös-Rényi (ER), Chung-Lu (CL), and Barabási-Albert (BA) models. x-axis denotes the degree. y-axis denotes the fraction of nodes. Plot is truncated; maximum values can be found in Table 3.2. 53

3.3 Comparison of graphs generated by the Erdös-Rényi (ER), Barabási-Albert (BA), Chung-Lu (CL), Configuration (CONF), and Configuration with Assortativity (CA) models with the $moderate_{50}$ (HCW) graph they model. Graph shows the mean number of people infected on a given day of the SIR simulation with transmissibility ($\rho = .0302$). x-axis represents the timestep for the simulation (days). y-axis gives the number of people in the "infected" state during that timestep. 57

3.4 Comparison of graphs generated by the Erdös-Rényi (ER), Barabási-Albert (BA), Chung-Lu (CL), Configuration (CONF), and Configuration with Assortativity (CA) models with the $moderate_{50}$ (HCW) graph they model. Graph shows the median number of people infected on a given day of the SIR simulation with transmissibility ($\rho = .0302$). x-axis represents the timestep for the simulation (days). y-axis gives the number of people in the "infected" state during that timestep. 58

3.5 Comparison of graphs generated by the Erdös-Rényi (ER), Barabási-Albert (BA), Chung-Lu (CL), Configuration (CONF), and Configuration with Assortativity (CA) models with the $moderate_{50}$ (HCW) graph they model. Graph shows the mean number of people infected on a given day of the SIR simulation with transmissibility $\rho = .0122$. x-axis represents the timestep for the simulation (days). y-axis gives the number of people in the "infected" state during that timestep. 61

3.6 Comparison of the HCW contact network and CONF graphs for transmissibility values in $\{.00763, 0.00794, 0.00824, 0.00854\}$. 64

3.7 Comparison of the HCW contact network and CONF graphs for transmissibility values in $\{.00763, 0.00794, 0.00824, 0.00854\}$. 66

3.8 Clustering coefficient distribution for the $sparse_{50}$, $moderate_{50}$, and $dense_{50}$ HCW contact networks. Vertices with degree less than 2 are ignored. . . 69

3.9 Vertex degree vs clustering coefficient correlation plot for the $sparse_{50}$, $moderate_{50}$, and $dense_{50}$ HCW contact networks. Vertex degrees are group in bins of size 10. The dense curve is truncated and extends out to 1240. 70

3.10 Two example graphs showing a joint degree-triangle distribution for a node u, where u has a single non-triangle forming edge $d_u = 1$, and is the corner of three triangles $t_u = 3$. (a) Configuration requiring 6 edges to 4 neighboring vertices and a clustering coefficient of .5. (b) Configuration requiring 10 edges to 7 neighboring vertices. 72

3.11 Clustering coefficient distribution for $sparse_{50}$ graph and corresponding graphs generated by the DEG algorithm. Vertices with degree less than 2 are ignored. 77

3.12 Vertex degree vs clustering coefficient correlation plot for $sparse_{50}$ graph and corresponding graphs generated by the DEG algorithm. Vertices are grouped into bins based on degree by groups of 5. 78

3.13 Comparison of $sparse_{50}$ graph and corresponding graphs generated by the CONF and CA random graph models and the DEG algorithm. Graph shows the average number of people infected on a given day of the SIR simulation with probability of transmission $\rho = .0122$. x-axis represents the timestep for the simulation (days). y-axis gives the number of people in the "infected" state during that timestep. 80

3.14 Comparison of $sparse_{50}$ graph and corresponding graphs generated by the CONF and CA random graph models and the DEG algorithm. Graph shows the average number of people infected on a given day of the SIR simulation with probability of transmission $\rho = .048$. x-axis represents the timestep for the simulation (days). y-axis gives the number of people in the "infected" state during that timestep. 81

3.15 Edgeswap operation on vertices y_1, y_2, z_1, z_2. 82

3.16 Clustering coefficient distribution for $sparse_{50}$ graph and corresponding graphs generated by the BKM model. Vertices with degree less than 2 are ignored. 86

3.17 Vertex degree vs clustering coefficient correlation plot for $sparse_{50}$ graph and corresponding graphs generated by the BKM model. Vertices are grouped into bins based on degree by groups of 5. 87

3.18 Comparison of $sparse_{50}$ graph with graphs generated using the BKM, CONF, and CA models with $\rho = 0.0122$. 88

3.19 Comparison of *sparse*$_{50}$ graph with graphs generated using the BKM, CONF, and CA models with $\rho = 0.048$. 89

3.20 Clustering coefficient distribution for *moderate*$_{50}$ graph and corresponding graphs generated by the SC model. Vertices with degree less than 2 are ignored. 93

3.21 Vertex degree vs clustering coefficient correlation plot for *moderate*$_{50}$ graph and corresponding graphs generated by the SC model. 94

3.22 Number of people infected on each day of an SIR simulation with low transmissibility $\rho = .0122$ for the *moderate*$_{50}$ graph and the Spatial-Clustering (SC), Configuration (CONF), and Configuration with Assortativity (CA) models. x-axis represents the timestep for the simulation (days). y-axis gives the number of people in the "infected" state during that timestep. . 95

3.23 Number of people infected on each day of an SIR simulation with low transmissibility $\rho = .122$ for the *moderate*$_{50}$ graph and the Spatial-Clustering (SC), Configuration (CONF), and Configuration with Assortativity (CA) models. x-axis represents the timestep for the simulation (days). y-axis gives the number of people in the "infected" state during that timestep. . 96

4.1 Simple graph of two vertices between two components of equal size. Vaccinating b alone gives really no improvement but vaccinating b along with a gives a major improvement by separating halves of the graph. For the *budgeted maximization problem* this demonstrates the objective function f violates the submodular property of "diminishing returns." 102

4.2 Instance of BV for which sum-of-squares partition problem is a poor proxy. 104

4.3 Comparison of experimental runs on the *moderate*$_1$ HCW contact networks for the greedy budgeted vertex cover algorithm and an upper bound calculated by solving the LP relaxation. The x-axis is truncates values that have the same y values. 109

4.4 Example graph showing that h_G+ is not submodular. 111

4.5 Bad Case for BMD. Suppose that $B = X - 1$ and $X = \frac{N}{3}$ so that each gray vertex has degree $\frac{2N}{3}$ and each black vertex has degree $\frac{N}{3}$. The optimal solution would be B black vertices resulting in reducing the maximum degree to $\frac{N}{3} + 1$. A greedy solution based on degree would pick B of the gray and the maximum degree would remain $\frac{2N}{3}$. The greedy algorithm is thus *at least* a 2 approximation. 112

4.6 Experimental comparison of the solution to k-PVC produced by Gandhi et al. [43] with the lower bound found by solving the corresponding LP. Points are annotated with the approximation ratio relative to the lower bound. 117

4.7 Comparison of approximate solutions for k-PVC used to solve instances of Max-BVC. The dashed line shows the result from **GreedyBVC** , the solid line shows the solution found by the primal-dual k-PVC algorithm given by Gandhi et al. [43], and the dotted line shows the lower-bound of the k-PVC problem based on the LP solution. 119

4.8 Comparison of experimental runs of the greedy solution to RMD an upper bound calculated by solving the LP relaxation on the $moderate_1$ graph. LP solutions are rounded up to the nearest integer. The x-axis denotes the integer k and the y-axis denotes the solutions size give as a fraction of the total number of vertices. 123

4.9 Example vaccination policies on a subgraph of an EMR contact network. (a) Small portion of the $sparse_1$ contact network (b) Results of vaccinating 50% of the population using the random policy. The result is still one single component. (c) Results of vaccinating 50% of the population using the degree-based policy. (d) Results of vaccinating 50% of the population using the distance-based policy. In both (c) and (d) the HCW contact network is “shattered” into many tiny components. 126

4.10 Effectiveness of vaccination policies on the $moderate_1$ HCW contact networks measured by the expected number of people infected as a result of simulation of an SIR disease diffusion simulation with peak transmission probability $p = .07$. The x-axis represents the budget of vaccinations as a fraction of the total population. The y-axis represents the expected fraction of the population infected based on experiments. 129

4.11 Effectiveness of vaccination policies on the $moderate_1$ HCW contact networks measured by the expected number of people infected as a result of simulation of an SIR disease diffusion simulation with peak transmission probability $p = .5$. The x-axis represents the budget of vaccinations as a fraction of the total population. The y-axis represents the expected fraction of the population infected based on experiments. 130

4.12 Effectiveness of five different vaccination policies on the $sparse_1$ contact network as measured by the size of the largest connected component in the unvaccinated network. The x-axis represents the percentage of people vaccinated. The y-axis represents the size of the largest connected component as a percentage of the size of the entire graph. 131

4.13 Effectiveness of five different vaccination policies on the *moderate*$_1$ contact network as measured by the size of the largest connected component in the unvaccinated network. The x-axis represents the percentage of people vaccinated. The y-axis represents the size of the largest connected component as a percentage of the size of the entire graph. 132

4.14 Effectiveness of five different vaccination policies on the *dense*$_1$ contact network as measured by the size of the largest connected component in the unvaccinated network. The x-axis represents the percentage of people vaccinated. The y-axis represents the size of the largest connected component as a percentage of the size of the entire graph. 133

4.15 Comparison of distance traveled, as measured by number of hops between consecutive logins, versus the degree of each vertex in the *moderate*$_1$ graph. 135

4.16 Effectiveness of vaccination policies on a "time-shifted" HCW contact network. The connectivity-based policies are generated from the *moderate*$_1$ HCW contact network. The mobility-based vaccination policies are generated using the EMR login data for the same time window $T = 1$. The plot shows the effectiveness of these policies on the "time-shifted" HCW contact network, *moderate*$_5$, measured by the expected number of people infected starting from a single infected individual chosen uniformly at random. The $T = 1$-network and the $T = 5$-network not only differ in edges, but also in the HCWs they contain as vertices. 137

4.17 Vaccination policies (a) degree and (b) distance for eight different *moderate* contact networks generated from four week periods spread throughout the 22 month period we have data for. Polices are evaluated by measuring the size of the largest connected component after vaccination. The x-axis represents the budget of vaccinations as a fraction of the total population. The y-axis represents the expected fraction of the population infected based on experiments. 138

4.18 Effectiveness of the degree-based, distance-based, k-PVC-based, and anonymous policies on the *moderate*$_1$ HCW contact networks measured by the expected number of people infected as a result of simulation of an SIR disease diffusion simulation with peak transmission probability $p = .5$. The x-axis represents the budget of vaccinations as a fraction of the total population. The y-axis represents the expected fraction of the population infected based on experiments. 139

4.19 Effectiveness of the degree-based, distance-based, RMD-based, and anonymous policies on the $moderate_1$ HCW contact networks measured by the expected number of people infected as a result of simulation of an SIR disease diffusion simulation with peak transmission probability $p = .5$. The x-axis represents the budget of vaccinations as a fraction of the total population. The y-axis represents the expected fraction of the population infected based on experiments. 140

4.20 Example graph of communities with high weight intra-community edges and low weight edges between communities. Thin lines represent low-weight edges and thick lines represent high-weight edges. Communities are circled by a dotted line. 141

4.21 Comparison of the effectiveness of the mincut-degree-based policy and degree-based policy. Policies are generate from the $moderate_1$ graph for time window $T = 1$, peak transmission probability $p = .07$, and $thresh =$ .3 144

4.22 Comparison of the effectiveness of the mincut-degree-based policy and degree-based policy. Policies are generate from the $moderate_1$ graph for time window $T = 1$, peak transmission probability $p = .1$, and $thresh = .3$ 145

4.23 Comparison of the effectiveness of the mincut-degree-based policy, and degree-based policy. Policies are generate from the $moderate_1$ graph for time window $T = 1$, peak transmission probability $p = .5$, and $thresh = .5$ 146

4.24 Trimming of boundary vertices as a result of the mincut percolation process.146

5.1 Summary statistics for all EIN traffic from February 1997 through May 2009 including the average (± standard deviation), minimum, and maximum number of mailings, number of posts, and number of users per thread.156

5.2 Data on classification of important threads. (a) percentage of threads whose number of mailings, number of posts, and number of participants are all within $thresh$ % of the corresponding maximum values of these characteristics. Since $T^+(thresh) \supseteq T^+(thresh')$ for $thresh > thresh'$, we obtain larger sets of important threads as we increase $thresh$ from 60 to 80. With $x = 60$, we pick up 18 (out of 2818) important threads, with $x = 70$, we pick up 47 (out of 2818) important threads, and with $x = 80$, we pick up 183 (out of 2818) important threads. (b) Aggregate statistics of important threads for different values of $thresh$. 158

5.3 A user-thread graph with red vertices (circles) denoting unimportant threads and blue vertices (squares) denoting important threads. For this instance with budget $B = 2$, **Greedy** will pick x and obtain a reward of 1, whereas the optimal solution consists of $\{y_1, y_2, \ldots, y_K\}$ for a reward of K. . . . 159

5.4 Plots showing the reward of solutions produced by **GreedyRatio** and **GreedyReward** with *thresh* = 80 and (a) $m = 2$ and (b) $m = 3$. The x-axis shows the budget and the y-axis shows reward. The black line shows the reward from **GreedyRatio** and the dashed line shows the reward from **GreedyReward**. The dotted lines mark points of interest, corresponding to 50%, 75%, and 100% coverage of important threads, discussed further in Table 5.2. 170

5.5 Map of selected users for full coverage at *thresh* = 80 and $m = 3$. 172

5.6 Plots of per-day post traffic on the EIN, averaged over seven days, in 2007. Results are from running **Greedy** with *thresh* = 80, full coverage, and (a) $m = 2$ and (b) $m = 3$. Gray levels show the total number of posts to the EIN. Black levels indicate the number of posts to marked threads only. 174

5.7 Plots comparing the performance of **Greedy** with a upperbound obtained by solving the LP relaxation of BMOC for *thresh* = 70 with (a) $m = 2$ and (b) $m = 3$. The x-axis is the budget and the y-axis is the reward. The black line shows the reward produced by the **Greedy** and the shaded line shows the upper bound obtained using the LP. 177

5.8 Plots showing the (small) improvement obtained by enhancing **Greedy** with "look-ahead;" all subsets of size at most 2 are considered as candidates in each iteration. Here *thresh* = 80 and $m = 2$. The x-axis is the budget and the y-axis is the reward. The black line shows the reward produced by **Greedy** without "look-ahead" and the dashed line shows the improvement due to "look-ahead." (a) shows the full plot and (b) shows the zoomed portion given by the box in (a). 179

CHAPTER 1
INTRODUCTION

There are many aspects of social networks that make them a useful tool in the field of epidemiology. And while there are a number of important developments to have come out of social network research relating to disease diffusion, there are also a number pitfalls when employing them on real-world social networks. In this work we address the problems of generating social networks that approximate close-proximity interaction from fine-grained spatiotemporal data, accurately modeling important aspects of these networks, and using these networks to improve decisions about vaccination policies and disease surveillance.

Social networks represent the relationships between individuals of a population. Online social networks (e.g., Facebook, Okrut, Baidu, etc.) replicate the structure of real-life relationships, which can be as strong as family members or as weak as acquaintances. This social structure can also be due to well-defined interactions such as communication over email [58], collaboration on a scientific publication [9], or the observed friendship between individuals [111].

Interest in social networks has a long history starting with famous experiments by Milgram [72] that suggested any two individuals were separated by "six degrees of separation" through acquaintances. In 1977 Zachary [111] observed that the split of a karate club was dictated by the structure of the friendship network. Barabasi et al. [9] use social networks to study the evolution of scientific collaboration. More recently Mas and Moretti [66] used social networks representing line-of-sight visibility of coworkers for cashiers in a grocery store to study peer effects on productivity. This is only a small sample of literature on social networks. A substantially broader treatment can be found in [79, 33].

Commonly, social networks are modeled as a graph $G = (V, E)$, with vertex set V, representing some population, and edge set E, representing relationships within this population. Modeling social networks in this way allows the use of network analysis techniques to understand these social relationships.

One of the major discoveries from analysis of these graph models of social networks was discovery of the "small-world" property by Watts and Strogatz [109]. The "small-world" property of social networks is a formalization of the conjecture by Milgram and suggests that any two individuals are separated by only a few connections, despite participating in a network that is only sparsely connected. Watts and Strogatz also showed that social networks tend to be highly clustered, with pairs of vertices sharing many of the same neighbors (i.e., your friends are also friends). Many social networks, such as the graph of the web, also evince a signature heavy-tailed degree distribution where a majority of vertices have a low degree and a few vertices have a very high degree [11]. More recently, Girvan and Newman [46] showed that a number of social networks have strong *community structure*. A more global property of clustering, strong community structure suggests that there are densely connected clusters (communities) of vertices and very few connections between clusters [81, 29].

Since disease spreads through close-proximity contact between individuals, social networks defined by physical proximity can provide valuable insights into how disease spreads. Motivated by the problem of reducing hospital acquired infections we examine four aspects of using social networks for epidemiology.

1.1 Hospital Acquired Infections

Nosocomial (hospital-acquired) infections are a major cause of morbidity and mortality in United States hospitals, causing up to an estimated 80,000 deaths a year [103, 52]. Nosocomial diarrhea due to *Clostridium difficile* is estimated to cost US hospitals over 1.1 billion dollars annually [62]. Usually hospital acquired infections enter a hospital through patients who, during their care, spread the disease to a healthcare worker (HCW). Infected HCWs then, in treating other patients and interacting with other HCWs, spread disease throughout the hospital. In the case of an outbreak of a nosocomial infection, the hospital can employ strategies for controlling the outbreak such as isolation [53], quarantining, sending infected HCWs home, or in certain circumstances, patient cohorting. These types of interventions can be modeled as optimization problems on social networks where the objective is to remove disease transmission pathways (edges) in order to separate sets of individuals (nodes). But for patient care, quarantine and isolation are problematic; isolation and quarantine severely degrade the level of patient care because HCWs are less willing to use additional precautions (respirators, gowns, gloves, etc.) required to treat patients in isolation.

Ideally it is better to simply prevent these outbreaks from happening. Some hospitals have instituted policies such as HCW uniform requirements, equipment sterilization, hand hygiene requirements, and vaccination requirements as a means of mitigating nosocomial infections. However, in most hospitals, hand hygiene and vaccination are still voluntary and both suffer from non-compliance. Aside from non-compliance, there are occasional vaccination shortages, notably in 2004 with influenza vaccinations [67] and, more recently, in 2009 with H1N1 vaccinations.

One way to deal with non-compliance and vaccination shortages is to “target” the right subset of HCWs for vaccination to protect the entire population. It has been shown that vaccinating the right subset of a population can ultimately lead to protection for the entire population, a phenomenon known as “herd immunity” [41, 6, 18]. As we will show, the problem of determining the “right” people to vaccinate can be modeled as an optimization problem. But solving optimization problems for mitigating hospital acquired infections requires that we first have an understanding of how disease spreads within the hospital environment.

1.2 Contact Network Epidemiology

The earliest models used to understand the diffusion of disease within a population were *compartmental* mathematical models such as SIR and its close relatives SEIR, MSEIR, and more recently SZR [50, 77]. These models track the size of compartments that divide the population by their “state”: being either *susceptible* (S), *infected* (I), or *recovered* (R). Compartmental models are based on the *mass-action principle* where the number of cases is proportional to the product of the number of infected and susceptible hosts. Recent research has shown that the mass-action principle can lead to inaccurate predictions, as demonstrated by the SARS outbreak in China [69, 92]. A fundamental problem of these compartmental models is the assumption of random mixing; infected individuals can spread disease to anyone else in the population. In reality we know that individuals have distinct contact patterns of whom they come into contact with.

Contact network epidemiology is a more powerful approach to studying disease diffusion that uses social contact networks to model close-proximity interactions which can lead to disease spread. For an outbreak of Severe Acute Respiratory Syndrome (SARS) in China, Meyers et al. [71] showed that contact network epidemiology could explain the inaccurate predictions by compartmental models which suggested a large scale epidemic. Eubank et al. [37] used census, land-use, and population-mobility data, to generate contact networks for the city of Portland, OR. Their experiments on these networks suggest that early detection is key for employing targeted vaccination. To study the spread of mycoplasma pneumoniae, Meyers et al. [70] modeled the contact network of a hospital based on the assumption that patients are confined to wards and disease is spread between wards by HCWs. They conclude that, given a uniform distribution of HCWs to wards, limiting the number of wards visited by a HCW and proper protection from airborne droplets are the best approaches to reducing spread of mycoplasma pneumoniae. Ueno and Masuda [105] simulated stochastic SIR simulations on social contact networks, based on patient records for a 129 bed Tokyo hospital with 500-600 employees, to study disease containment strategies. Under a number of assumptions about the contact patterns of HCWs and patients, they conclude that physicians should be prioritized for vaccination. Most recently Polgreen et al. [98] use contact networks gathered from observational data to show the importance of contact structure for informing vaccination interventions.

One of the problems of research in contact network epidemiology is the lack of reliable data from which to infer contact networks that are epidemiologically relevant. There is now considerable research on the structure of online social networks (see for example [4, 73, 56, 17]), but such online social networks are not always epidemiologically relevant and may be structurally very different from networks of HCWs induced by spatiotemporal proximity. The contact networks used by Meyers et al. [70], Ueno and Masada [105], and Polgreen et al. [98] are relatively small and constructed on the basis of limited data, taking a rather coarse view of time and the hospital space in which interactions take place. As a result, these approaches result in contact networks that are either highly structured (e.g., consisting of a clique for each ward or unit) or drawn at random from simple probability distributions. Neither of these types of networks seem representative of the complexity of interactions that occur in real hospital settings.

In Chapter 2, we present a comprehensive approach to constructing *HCW contact networks* in a large hospital setting via the use of electronic medical records (EMR). We apply this approach at the University of Iowa Hospitals and Clinics (UIHC), a 3.2 million square foot facility with 700 beds and about 8,000 HCWs. Using a data set of over 19.8 million EMR logins spanning more than 21 months (Sept 1, 2006 through June 21, 2008), we construct 9,000 different HCW contact networks for the UIHC. These contact networks serve as realistic proxies for patterns of actual HCW contacts and provide some of the most detailed views, as yet, of contacts among hospital-based HCWs. Analysis of these contact networks reveals that despite spatial and job-related constraints on HCW movement and interactions, there is a surprising structural similarity between the HCW contact networks we generate and social networks that arise in other settings (e.g., movie or scientific collaborations, on-line friendships, etc. [4, 9, 58, 109]).

1.3 Random Graph Models for Contact Networks

Contact network structure plays a key role in how disease will spread on a network [69, 84, 109]. With the exponential growth in the size of networks being studied – compare the karate club graph of Zachary [111] in 1977 with 34 vertices and 78 edges to the web graph studied by Broder [19] in 2000 with 200 million vertices and over 1.5 billion edges – it is no long possible to visually analyze structure. Modeling networks as random graphs is a way to focus on the important structure aspects while abstracting away the unimportant features. In addition, random graph models can be used to predict properties of the class of networks represented by the model. Thus, finding random graph models that *accurately* model real-life instances of social networks are essential to the success of contact network epidemiology.

One of the major fallouts from the work by Watts and Strogatz [109] was realization that real-world social networks have distinct structural differences from the Erdös-Rényi random graphs. Thus, a lot of recent work has focused on the development of graph models that more accurately represent social networks. [61, 12, 89, 83, 8, 85, 87]. Development of accurate random graph models as primarily focused on replicating the degree distribution of vertices, [14, 27, 75, 89]. Barabási and Albert [13, 12] proposed a model for graphs with a power-law degree distribution in order to capture the structure of the web graph. Considerable focus has been paid to random graphs with an explicit degree sequence based on a model by Bender and Canfield [14]. Molloy and Reed [75, 76] showed that there exists edge density threshold for the emergence of a giant component. This work was extended by Newman [84] and Meyers [69] who provide calculations for expected disease outbreak size. Chung and Lu [27] propose a model for graphs with an expected degree distribution which was extended by Eubank et al. [38] to model people and the locations they visit for the city of Portland, OR. More recently, Newman [82] has proposed new graph models for graphs with given degree sequence and neighbor correlations. Bansal and Meyers [8] and Newman [87] have both introduced graph models that capture degree sequence and vertex clustering.

In Chapter 3 we consider a number of candidate random graph models to describe our HCW contact networks. Our results show that simple random graph models that only pay attention to local structure (i.e., mean degree and degree distribution) fail to capture epidemiologically relevant aspects of the HCW contact networks, and thus may be poor models for real world social networks in general. Moreover we show compelling evidence that, for HCW contact networks, the correlation between degrees of adjacent (neighboring) vertices plays an important role in disease diffusion. Finally, experiments on random graph models with clustering suggest that local clustering does have a profound effect on the spread of disease.

1.4 Vaccination Policies

Vaccination is an easy way to prevent hospital acquired infections [100, 110]. Yet, non-compliance and vaccination shortages [67] remain a problem. Recent research suggests that employing “targeted” vaccination strategies is a viable option for protecting the entire population in spite of these problems [41, 6, 18]. However, the effectiveness of targeted vaccination strategies require knowledge of *who* to vaccinate in order to minimize the spread of disease. Social contact networks and disease diffusion models provide valuable insight into who these “key” individuals are.

If we suppose that disease diffusion is a dynamical process over a social contact network $G = (V, E)$, then there are two natural optimization problems that fall out. The first problem, which we call *budgeted vaccination*, is to find key individuals to target with vaccination in the case of vaccination shortages. That is, given a budget b of vaccinations, find a size-b subset $V' \subseteq V$ to vaccinate such that the number of vertices infected as a result of disease diffusion on $G \setminus V'$, the graph resulting from the removal of V' from G, is minimized. The second problem, which we call *restricted disease*, is to minimize the number of individuals that need to be vaccinated in order to “restrict” disease spread to a given size of the population. More precisely, given integer budget $k < |V|$, find a minimum size subset of vertices $V' \subseteq V$ whose removal from G reduces the expected number of people infected as a result of a single infected individual is less than k.

Kempe et al. [54] have considered the *influence maximization* problem, similar to budgeted vaccination, of finding a size-k subset S of individuals to “infect” with an idea so as to maximize the number of individuals influenced as a result of a “word-of-mouth” diffusion process over a social network. Supposing that the number of influenced individuals is given by an oracle function f, they show that, for a number of diffusion models, the simple greedy algorithm that continually adds to S the individual that maximizes $f(S)$ provides a near-optimal solution. Goyal et al. [48] consider the dual problem of influence maximization, similar to restricted disease, of finding the minimum size set of initial individuals to implant with an idea so that the number of influenced individuals is above a given threshold. They show that, despite its similarity to influence maximization, the problem is quite hard.

There has also been recent work done on game-theoretical aspects of vaccination [7, 26]. Aspnes et al. [7] consider a game-theoretic model of vaccination where individuals can choose to get vaccinated or not. Their main result suggests that, left to their own devices, selfish individuals make decisions that are bad for the whole. Under the assumption that disease spreads in a worst case fashion to all unvaccinated individuals in the contact network, Aspnes et al. [7] introduce the *sum-of-squares partition problem.* Given a graph G of n nodes and budget B, the sum-of-squares partition problem is to find a set of B vertices whose removal from G minimizes the size of the largest connected component. Aspnes et al. [7] show that an $O(\log^{1.5}(n))$-approximation can be achieved in polynomial time.

To solve problems like the budgeted vaccination and restricted disease problems, many researchers make significant assumptions about the behavior of disease diffusion to make the problems more mathematically tractable. In Chapter 4 we consider a number of graph optimization problems to act as surrogates for solutions to the budgeted vaccination and restricted disease problems. We show that the quality of the solutions provided by the surrogate optimization problems lead to solutions that are poor solutions to the budgeted vaccination and restricted disease problems in general. However, experiments on our HCW contact networks suggest that they may be perfect candidates for these surrogate problems. As a consequence, a simple greedy algorithm that picks the most well connected individuals (i.e., those having high degree) provides a near optimal solution to budgeted vaccination and restricted disease problems. Finally we compare a number of heuristic policies for vaccination on the HCW contact networks we generate. Our results show evidence that vaccinating the most "mobile" individuals may be an effective vaccination strategy.

1.5 Disease Surveillance

Disease surveillance and early detection of outbreaks may be one of the most important disease control strategies [36, 95]. The Emerging Infections Network (EIN) is a network of clinical infectious disease specialists created with the goal of assisting the CDC and other public health authorities with surveillance of emerging infectious diseases and related phenomena (new treatment protocols, possible side effects of new vaccines, etc.). To achieve its goal, the EIN maintains a private listserv of over 1400 infectious disease specialists, CDC investigators, and public health officials. Since its inception, the EIN listserv has served over 2800 discussions on the identification of new infectious diseases, treatments, and policy implications. Identifying important topics of discussion on the EIN is currently ad hoc, done a list administrator reading all discussions. There is significant interest in improving the accuracy and timeliness with which this important information is identified so that it can be distributed to the CDC and other healthcare organizations.

There are a number of approaches that have been taken previously to improve disease surveillance methods. Polgreen et al. [95] considered the problem of finding optimal placement to increase coverage of an influence surveillance network. They show that maximum coverage models can greatly increase the coverage level for the state of Iowa. Polgreen et al. [97] have considered the use of healthcare prediction markets, emerging from economics [42], to give timely predictions based on healthcare related forecasting. Other recent work has focused on using the collective wisdom of crowds to track disease outbreaks using search engine queries [45, 93]. During the H1N1 outbreak in 2009, a number of projects considered the use of Twitter posts to track the spread of the infection [101].

Our solution to improve disease surveillance is to develop a simple procedure for identifying discussions on the EIN that have the potential to become “important,” and ignore threads that will remain “unimportant.” To solve this problem we leverage the social network of individuals and their participation in threads, based on historical EIN data, and identify a set of “bellwether” users who typically participate in the early stages of many important threads, but are involved in very few unimportant threads. If we are able to identify such "bellwether" users, then tracking these users can quickly point people who make policies to emerging important threads that are in their early stages of evolution, without inundating them with irrelevant information.

A similar problem has been considered by Leskovec et al. [65] for the placement of contamination sensors in a water distribution network. They show that this approach can be extended to the unseemingly similar problem of selecting a set of blogs to monitor so as to catch the maximum number of important news stories. More recently El-Arini et al. [34] considered a similar problem of providing personalized monitoring of the blogosphere, tailored to individual users. In both cases these problems can be formalized as instances of *submodular maximization problems* that have a long history starting with Nemhauser et al. in 1978 [78]. For set U the function $f : 2^U \to \mathbb{R}^+$ is said to be submodular if it exhibits the property of "diminishing returns": $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ for all $A, B \subseteq S$. Nemhauser et al. [78] consider the problem of finding a subset $S \subseteq U$ of given cardinality which maximizes $f(S)$ and show that a simple greedy algorithm that continually adds to S the element $u \in U$ which maximizes $f(S \cup \{u\}) - f(S)$ provides a near-optimal solution. Khuller et al. [55] extend this result to a problem *budgeted maximum coverage* where each element $u \in U$ has associated cost c_u and the objective is to find subset $S \subset U$ such that $f(U)$ is maximized and $\sum_{u \in S} c_u \leq b$ for some budget b. More advanced constraints have also been shown to have near-optimal approximations [21, 20].

In Chapter 5 we show that the problem of determining "bellwether" users on the EIN can also be modeled as a generalization of budgeted maximization problem that we call *budgeted maximization with overlapping costs* (BMOC). Due to its unique cost structure, BMOC is fundamentally different than those applied previously and thus simple greedy approaches do not work. In fact, a simple reduction to the *densest k-subhypergraph* [49] problem shows that BMOC problem is very hard in general. However, by identifying a possible feature of the EIN social network, which we call the *overlap condition*, we show that for certain instances of BMOC a simple greedy algorithm does provide a near-optimal (constant-factor) approximation. Finally, experimental runs of the greedy algorithm provide strong evidence that the EIN data exhibits this *overlap condition* and thus solutions obtained are very close to optimal.

CHAPTER 2
GENERATING HCW CONTACT NETWORKS

In this chapter we introduce a graph representation of spatial information about the University of Iowa Hospitals and Clinics (UIHC) – a 3.2 million square foot facility with 700 beds and about 8000 healthcare workers – and a set of over 19.8 million de-identified healthcare worker activity logs, based on login records for an electronic medical records (EMR) system. Using these data we introduce a comprehensive method for constructing a healthcare worker (HCW) contact networks that serve as proxies for contact patterns between HCWs. Analysis of the constructed contact networks reveals that despite spatial and job-related constraints on healthcare worker movement and interactions, there is a surprising structural similarity between the healthcare contact networks we generate and social networks that arise in other settings (e.g., movie or scientific collaborations, on-line friendships, etc.).

2.1 Constructing HCW Contact Networks

The biggest obstacle to using contact networks in epidemiology is the absence of reliable data from which to infer contact networks that make epidemiological sense. There is now considerable research on the structure of online social networks (see for example, [4, 73, 56]) and on how information travels through these networks [57]. But such online social networks are not always epidemiologically relevant, as they are not based on physical contact, and may be structurally very different from networks induced by spatial proximity.

login date & time	logout date & time	device	location	userID	position & dept.
2006-09-01, 0:00:00.40	2006-09-01, 0:24:17.29			SKR925	STAFF NURSE I, NURSING
2006-09-01, 0:00:00.43	2006-09-01, 0:00:21.76	M95089	JPP 6750	SLB565	STAFF NURSE II, NURSING
2006-09-01, 0:00:01.23	2006-09-01, 0:03:55.21			HNH286	STAFF NURSE II, NURSING
2006-09-01, 0:00:02.29	2006-09-01, 0:00:14.81	MA1458	RCP 1100	K920	HOUSE STAFF III, NEUROLOGY
2006-09-01, 0:00:02.54	9-1-06, 0:46:37.82	B71118	RCP 1047	M811	HOUSE STAFF I, ETC

Figure 2.1: The first five of approximately 19.8 million EMR login records. The `UserIDs` are all de-identified, although each de-identified user has an associated `position & dept` field. The `Device` field provides computer IDs with associated `Location` information. The `Location` field specifies rooms in the UIHC (e.g., `RCP 1100` is room number 1100 in the Roy Carver Pavilion of the hospital). Note that some of the records are missing the `Device` field, rendering them unusable for contact graph construction. needed for contact network construction, still leaving about 11.7 million usable records.

In general, close physical proximity or contact with a common physical surface (e.g., door knob or keyboard) is necessary for the spread of an infection. However, data representing spatial proximity among members of a sizable population are hard to come by. We deal with this problem by using EMR login data. Employees of the UIHC use the EMR system multiple times over the course of a day and each "login event" is recorded (see Figure 2.1). The EMR system uses an automatic logout system, due to HIPPA rules, so login times correspond very closely to the times when a HCW is physically at the login terminal.

Each event is logged by date, time, anonymized user ID, and location, providing a rich context from which to infer contact and movement. The aggregate characteristics of these data given in Figure 2.2 show not only the large number of healthcare workers (15,595) represented in this data, but also their diversity (80 departments, 404 job titles). The 4,379 locations of the computers are well spread out around the hospital. Most computers are located inside out-patient rooms, in clusters just outside groups of in-patient rooms, at nurses' stations, at the desks of unit

records	days	users	depts	positions	devices	locations
19.8 million	660	15,595	80	404	17,522	4,379

Figure 2.2: Table showing the size and other aggregate characteristics of the EMR login data.

clerks, and in doctors' offices. This distribution of computers implies that healthcare workers do not have to travel just to login to the EMR system and therefore strongly suggests that locations of healthcare worker logins are well correlated with their daily activities.

We construct the contact networks in two steps. In the first step, we construct a detailed discrete spatial model of the UIHC space that allows us to determine spatial proximity of login locations; this information is critical to the contact network construction. The hospital model also allows us to estimate the mobility of healthcare workers, a measure that we use to inform vaccination policies. In the second step, we parse EMR login data and construct various contact networks based on several network generation parameters. These two steps are described in the next two subsections.

2.1.1 The Hospital Graph

Eleven buildings or permanent additions connected by corridors make up the main UIHC complex, which contains 3.2 million gross square feet and covers an area of about 13.8 acres. The straight-line distance from the northern end of the complex to the southern end is about 1,600 feet (roughly 0.3 miles or 3.6 blocks). The actual walking distance through the corridor system is about 2,000 feet.

We model this space as a graph whose vertices represent rooms and whose edges represent adjacencies between rooms. Corridors and large spaces (e.g., atriums and cafeterias) are partitioned into smaller spaces so that each vertex represents an area of about the same size. The hospital graph allows us to approximate walking distances in the hospital by hop distances in the graph (see Figures 2.3 and 2.4). This discretization allows us to easily compute various distance-based characteristics of the hospital. The hospital graph was constructed manually using data from two sources provided by the UIHC: (i) a spreadsheet containing most of the rooms in the hospital along with their names, floor numbers, area in square feet, and purpose, (ii) architectural CAD drawings that showed blueprints of each of the floors. We manually (and painstakingly!) extracted room adjacencies from the CAD drawings and through a combination of manual and algorithmic efforts, we were also able to extract approximate 3-dimensional coordinates for all the vertices in the hospital graph (see Figure 2.5).

The graph we constructed has 19,554 vertices and 23,566 edges. Given the 3.2 million square foot area of the hospital, this implies that on average each vertex corresponds to 163.65 square feet in area (i.e., a 12.5 foot × 12.5 foot room). Due to discrepancies between the hospital room spreadsheet and the hospital CAD drawings, the graph has a small number of small connected components and one "giant" component with 18,961 vertices and 23,442 edges. We delete the small components and take this giant component to be the *hospital graph* (see Figure 2.6). The hospital graph essentially overlays a metric space (induced by pairwise hop-distances between

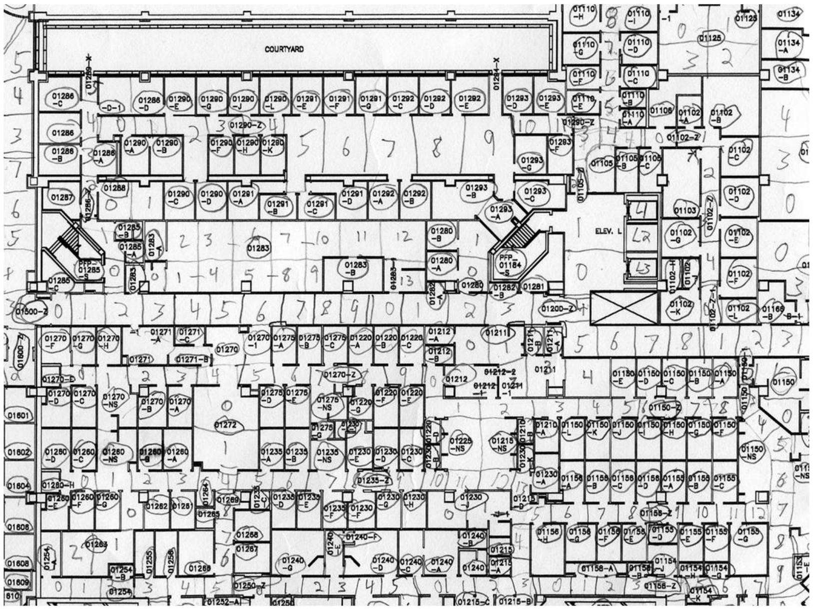


Figure 2.3: A CAD drawing fragment for the basement (floor 0) of the hospital, showing how it was marked up by hand in order to break up corridors into segments that were approximately room-sized.

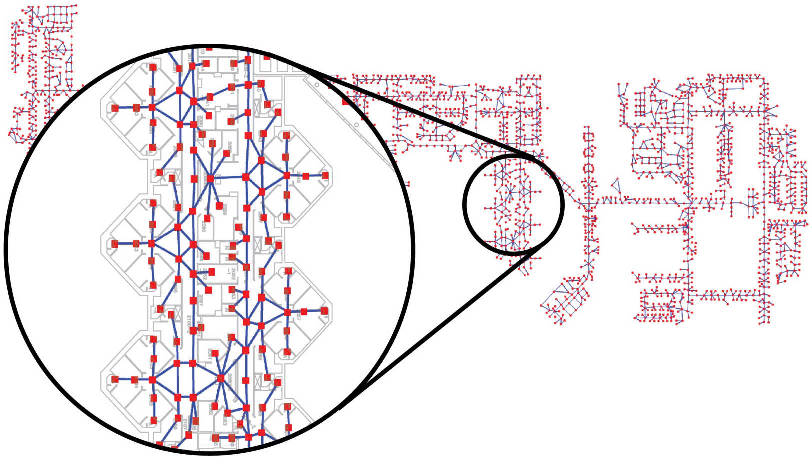


Figure 2.4: A small portion of the hospital graph, corresponding to the second floor of the UIHC. The inset makes clear how each room or corridor segment is represented by a vertex, connected by edges to adjacent rooms or corridor segments. This particular image was produced by superimposing the graph onto a CAD drawing of the floor plan.

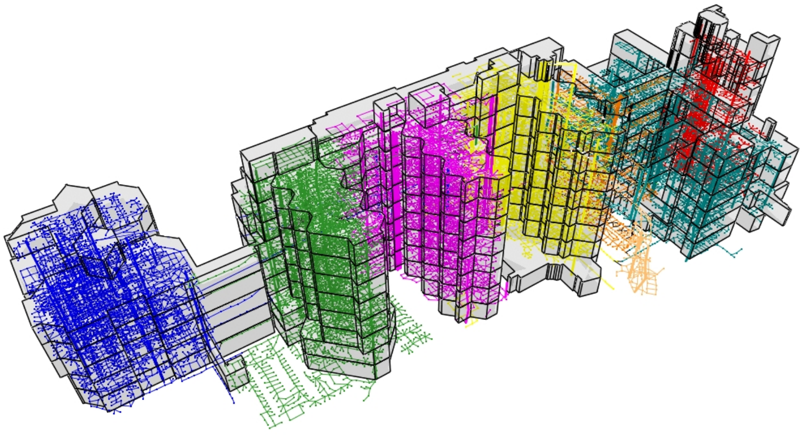


Figure 2.5: This picture shows the entire hospital graph superimposed on a 3-dimensional architectural drawing of the hospital. The vertices are colored according their building designator.

hospital vertices) on the UIHC facility and plays a critical role in a number of aspects of our work where spatial proximity (or lack thereof) is important.

Vertices	Edges	Mean Degree	Diameter	Mean Path Length
18,961	23,442	1.236	137	102.39

Figure 2.6: Basic characteristics of the hospital graph. This graph has an average degree of 1.236, which is consistent with our observation that most rooms have degree one or two because they connect only to a corridor or to a corridor and a bathroom.

2.1.2 Extracting contacts

Fix a time window T that corresponds to a contiguous sequence of days during the time period between 2006-09-01 and 2008-06-21, that we have login data for. For example, T could be the 4 week period starting on 2006-09-03 and ending on 2006-09-30. Let V^T denote the set of users who have logged into the EMR system at least once during time window T. Fix parameters $d \geq 0$ and $t \geq 0$. Each healthcare worker $u \in V^T$ has a set L_u of login sessions, where each login session $I \in L_u$ is defined by its *start time* $s(I)$, its *end time* $e(I)$, and its location or *placement* $p(I)$. Two healthcare workers $u, v \in V^T$ are connected by an edge if for some login sessions $I \in L_u$ and $I' \in L_v$, the distance in the hospital graph between $p(I)$ and $p(I')$ is at most d hops and the time interval $[s(I)-t, e(I)+t]$ intersects the time interval $[s(I'), e(I')]$. In other words, u and v are connected by an edge if their login sessions occur within t time units of each other and within d hops of each other in the hospital graph. The edge $\{u, v\}$ is assigned an edge-weight $w(u, v)$ that is the number of distinct login sessions I and I' that satisfy the above conditions. Thus $w(u, v)$ represents the number of distinct contacts between u and v, within the specified time window T, as indicated by their login records. Varying the values of d and t allows us to consider alternate notions of when a contact occurs. Specifically, as d and t increase, we essentially "loosen" the definition of a contact, thus producing denser contact graphs. The d and t values for which we have constructed various *healthcare worker contact networks* – a *HCW contact network*, in short – are given in Figure 2.7). For our discussion we focus use the names $\texttt{sparse}_i$, $\texttt{moderate}_i$, and $\texttt{dense}_i$ to denote the HCW contact networks with parameters $(d = 1, t = 0, T = i)$, $(d = 3, t = 15, T = i)$, and $(d = 5, t = 30, T = i)$ respectively. Example subgraphs for the $sparse_1$, $moderate_1$, and $dense_1$ graphs are given in Figure 2.8.

	Contact Network Generation Parameters	Possible Values
$d =$	maximum hop-distance between pairs of login locations	$0, 1, 2, 3, 5$
$t =$	maximum time (in minutes) between pairs of logins	$0, 5, 10, 15, 30$
$T =$	a 4-week time window completely within the period 9-1-06 and 6-1-08	$000, 001, 002, \ldots, 089$ 000 starts on 2006-09-01, 001 starts on 2006-09-08, etc.

Figure 2.7: The different parameters and their possible values that we use for generating healthcare worker contact networks. With 5 values for d, 5 for t, and 90 for T, all independently chosen, we obtain over 2,250 different healthcare contact networks.

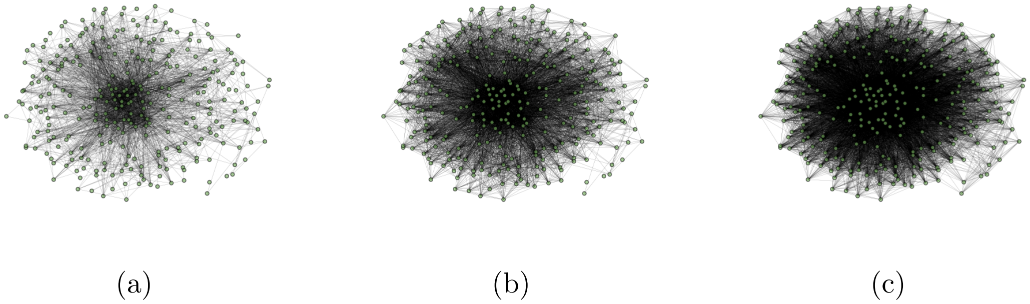


Figure 2.8: Small section of HCW contact networks generated based on EMR login data using different definitions of a contact. (a) Contact graph generated with $d = 1, t = 0$. (b) Contact graph generated with $d = 3, t = 15$. (c) Contact graph generated with $d = 5, t = 30$.

2.1.3 HCW contact networks: discussion

The high resolution of our EMR login data allows us to extract from it encounters between pairs of healthcare workers who have "weak ties." This might include pairs of healthcare workers who work together only occasionally, e.g., to deal with an unusual patient. Contacts between such pairs of healthcare workers could not have been easily predicted by static, coarse-grained data, e.g., department affiliations or job types. "Weak ties" influence the structure of contact networks in critical ways, significantly influencing the spread of disease. The high resolution of our data also shows a great deal of diversity of movement and interactions among healthcare workers within the same department and within the same job type. This is another important outcome of our approach discussed with more detail in section 2.3.

We are aware of several problems with using HCW contact networks as a proxy for patterns of actual healthcare worker contacts. These include the complete absence of patients and certain categories of healthcare workers such as janitors and transporters. Another problem is that certain healthcare worker behaviors that may introduce a systematic bias in the EMR login data. For example, typically healthcare workers visit patients in small groups during rounds and designate the junior-most member as the person in-charge of updating the EMR system. This may cause a relative absence of senior staff logins in the EMR login data, even though the senior staff may be moving around the hospital and interacting with patients and other healthcare workers as much as the junior staff.

We aim to address these problems in the future using a combination of new data gathering techniques (e.g, having HCWs and patients wear wireless “badges” that will record contacts) and further analysis of available data (e.g., patient admission and discharge data, out-patient load data, etc.). Recently we have made progress implementing the wireless “badge” approach at the UIHC to detect the proximity of individuals [31] and automate monitoring of HCW hand hygiene [96].

Computers	People	average person degree	average computer degree
4,861	6,875	14.13 (± std. dev. 25.832)	9.99 (± std. dev. 12.562)

Figure 2.9: These statistics show that the computers-people graph is relatively sparse However, both person-degrees and computer-degrees show a large standard deviation raising the possibility of a few heavily used computers and a few highly mobile healthcare workers.

2.2 Computers-People Graph

An alternate graph-theoretic view that explicitly shows the interactions between healthcare workers and computers is the *computers-people graph* (see Figure 2.9 and Figure 2.10). The computers-people graph is a bipartite graph where one part consists of healthcare workers and the other consists of computers. Roughly speaking, an edge is placed between a healthcare worker and a computer if the computer was used by that individual during a particular time window based on the EMR login data. More precisely, fix a time window T and let U^T be the set of computers which had at least one login during time period T, and V^T be the set of healthcare workers that logged into the EMR system at least once during time period T. Each computer $u \in U^T$ and each user $v \in V^T$ is connected by an edge $\{u, v\}$ if v has logged into u at least once during time period T. The edge $\{u, v\}$ is assigned an edge-weight $w(u, v)$ equal to the number of times v has logged into u during time window T.

The computers-people graph encodes a variety of useful information. For example, the degree of each healthcare worker in this graph captures the "login-heterogeneity" patterns of a healthcare worker's access the EMR. In Chapter 4 we evaluate a vaccination policy in which healthcare workers with highest degree in the

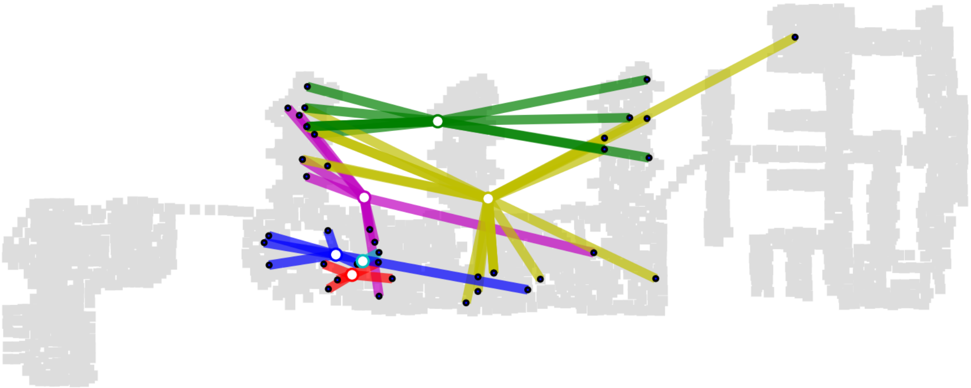


Figure 2.10: Small portion of a computers-people graph for a 4-week period. Computers are marked by black dots and are shown in their actual location on the 4th floor of the UIHC. Healthcare workers are shown as white dots and the edges connect healthcare workers to the computers they have logged into during the 4-week time window.

computers-people graph are vaccinated first and show that this policy also performs much better than the policy that picks healthcare workers uniformly at random.

2.3 Analysis of HCW Contact Networks

One of the premises of contact network epidemiology is that individual contact patterns can be quite diverse and this diversity substantially affects the spread of infectious diseases. In their seminal paper, Watts and Strogatz [109] point out that "real world" networks such as movie collaboration networks or the power grid network in the Western United States have structural characteristics that are quite different from those possessed by the well-known Erdös-Renyi random graph model [35]. The Erdös-Renyi random graph model, denoted $G(n, p)$, is an n-vertex graph in which each pair of vertices u and v are independently connected by an edge with probability

p. Thus the Erdös-Renyi model essentially takes all vertices and edges to be identical, at least in a probabilistic sense. But, as observed by Watts and Strogatz [109] edges and vertices in “real world” networks exhibit a lot of diversity. This also is the case for the HCW contact networks that we generate.

Since the work of Watts and Strogatz [109], research on modeling social networks has taken off, initially spurred by the growth of the web and now by the widespread use of online social networking sites such as Facebook, LinkedIn, MySpace, Wikipedia, digg, del.icio.us, and even YouTube, and Flickr. This line of research has focused on a number of structural features of social networks. We focus on three features that seem most relevant from an epidemiological point of view: (i) *degree distribution*, (ii) *small world property*, and (iii) *community structure*. Figure 2.11 shows statistics pertaining to these features for three representative HCW contact networks.

2.3.1 Degree Distributions

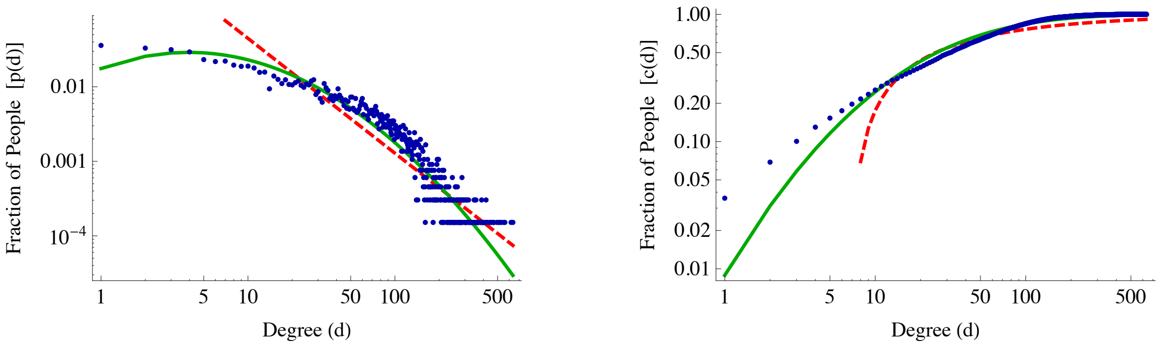
It is well-known that the degree distribution of the Erdös-Renyi random graph $G(n, p)$ is binomial (Poisson, in an asymptotic sense). The binomial distribution is sharply concentrated about its mean yielding a small standard deviation. In all three cases ($sparse_1$, $moderate_1$, and $dense_1$), the standard deviation of the degree distribution of the login contact graphs is much larger than that of the Erdös-Renyi graphs (see rows corresponding to σ and σ_{rand}), indicating a degree distribution that is much more dispersed than the binomial distribution. Also, the fraction of people

	sparse$_1$	*moderate*$_1$	*dense*$_1$
n (num. vertices)	6,875	6,875	6,875
m (num. edges)	82,199	174,739	332,766
$\langle k \rangle$ (mean degree)	23.91	50.83	96.8
k_{max} (max. degree)	321	635	1,115
σ (std. dev. degree dist.)	32.84	62.86	113.877
σ_{rand} (std. dev. degree dist. $G(n, p)$)	4.90	7.06	9.77
cc (clust. coeff.)	0.3109	0.3906	0.4379
cc_{rand} (clust. coeff. $G(n, p)$)	0.003516	0.007476	0.01414
c (num. components)	873	293	144
c_{rand} (num. components $G(n, p)$)	1	1	1
n_{giant} (num. vertices giant comp.)	5,838 (84.92%)	6,547 (95.23%)	6,702 (97.48%)
m_{giant} (num. edges giant comp.)	81,935 (99.68%)	174,687 (99.97%)	332,717 (99.98%)
$diam$ (diam. giant comp.)	11	13	12
$\langle \ell \rangle$ (ave. path len. giant comp.)	3.592	3.131	2.746

Figure 2.11: Basic structural features of a *sparse*$_1$ ($d = 1$, $t = 0$), *moderate*$_1$ ($d = 3$, $t = 15$), and *dense*$_1$ ($d = 5$, $t = 30$) instance of the HCW contact network are shown here. Note that the dense graph is only dense relative to the sparse graph; the average degree of even the dense graph is less than 1% of the graph size. For all three graphs we use time window $T = 001$, i.e., a 4-week time window starting at the second week of our EMR login data. For comparison, the corresponding statistics for Erdös-Renyi random graphs with same size (n) and same mean degree ($\langle k \rangle$) are also provided.

whose degree is no greater than the average vertex degree is 66.89% (*sparse*$_1$), 64.97% (*moderate*$_1$), and 64.00% (*dense*$_1$), pointing to a heavy tail in all of these cases. The plots in Figure 2.12 confirm this. Figure 2.12(a) shows the log-log plot of the degree distribution of the *moderate*$_1$ HCW contact network, indicating quite clearly that the distribution is heavy-tailed, covering close to three orders of magnitude and indicating a high level of heterogeneity among healthcare workers. This has important implications for infection control: if indeed a few people have lots of contacts, then it seems natural to try and target this group for vaccination.

We have also analyzed the degree distributions of the HCW contact networks with the aim of determining how well the popular heavy-tailed *power-law distribution* and *log-normal distribution* [74] fit the observed degree distributions. Figure 2.12



(a) (b)

Figure 2.12: Degree distribution of the $moderate_1$ HCW contact network and max likelihood power-law (dashed line) and log-normal (light curve) fits. (a) log-log plot; each point $(d, p(d))$ represents the fraction $p(d)$ of healthcare workers with degree d in the moderate HCW contact network. (b) cumulative form; each point $(d, c(d))$ represents the fraction $c(d)$ of healthcare workers with degree at most d) is shown here.

shows the fits visually for the $moderate_1$ graph and both fits seem reasonable, especially when viewing the cumulative density function (cdf) plot (Figure 2.12(b)), with the log-normal seeming to be a better fit. The plots for the sparse and dense case are similar. For both power-law and log-normal fits, we select the optimal parameter values for the distributions using a version of the maximum likelihood estimation method suggested by Clauset et al. [28]. Notwithstanding the plots, the p-values from a Kolmogorov-Smirnov test (following the approach of Clauset et al. [28]) indicate that while neither power-law nor log-normal are particularly good fits for any of the degree distributions, the log-normal distribution seems to be a marginally better fit than power-law, at least for the $sparse_1$ and $moderate_1$ graphs.

2.3.2 Community Structure

The *clustering coefficient* of a graph can be defined as follows. Let d be the degree of a vertex v. The maximum number of edges possible between neighbors of v is $\Delta := d(d-1)/2$. The *clustering coefficient of a vertex* v, denoted $cc(v)$ is the ratio of the actual number of edges between neighbors of v to Δ. The clustering coefficient of a graph is the average of $cc(v)$ over all vertices v. In social networks, $cc(v)$ measures the extent to which people that v comes into contact with, also come into contact with each other. In other words, $cc(v)$ measures the extent to which the neighborhood of v forms a community. One of the observations that motivated the work of Watts and Strogatz [109] is that the real world networks they examined had clustering coefficients that were orders of magnitude larger than the clustering coefficient of the comparable Erdös-Renyi graphs. A comparison between cc and cc_{rand} in Figure 2.11 shows that this is the case for the HCW contact networks as well.

Communities need not be restricted to neighborhoods in a network and an additional property of social networks that has attracted a lot of attention is their *community structure* [46, 81]. Informally speaking, a graph is said to have a strong community structure if it can be partitioned into groups of nodes that are densely connected with very few edges between groups. This structural feature can be measured in many different ways, the particular measure we consider, defined by Newman and Girvan [81], is called *modularity*.

Defined loosely, the *modularity* of a given vertex-partition of a graph (i.e., the

	$sparse_1$	$moderate_1$	$dense_1$
job class	.08	.05	.03
department	.21	.15	.12
spatial	.366	.312	.272
`maxQ`	.50	.38	.33

Figure 2.13: Modularity values for partitions of $sparse_1$, $moderate_1$, and $dense_1$ graphs obtained via different methods. The modularity of the partition induced by "job class" falls well below the 0.3 threshold for a strong community structure, mentioned by Newman [80]. The last row shows modularity values for partitions obtained by using a greedy clustering algorithm, called **`maxQ`**, due to Clauset et al. [29]

graph's community structure) is the fraction of intra-community edges compared to the fraction of intra-community edges that the same node partition would have with a uniform random assignment of edges (see [81] for a precise definition). Modularity values upwards of 0.3 are said to indicate strong community structure [80]. Healthcare workers can be naturally partitioned by job class or by department. As shown in Figure 2.13, community structure induced in this way does not appear to be particularly strong. This is not surprising because healthcare workers in the same job class (e.g., nurses) are widely dispersed across multiple departments and departments are often composed of spatially dispersed units. One can do somewhat better by using spatial attributes. Specifically, for each healthcare worker u, define a *home location* $H(u)$ as the location of the computer in the hospital graph that u logs into most often. This maps each healthcare worker onto a vertex in the hospital graph and moreover establishes a metric space on the set of healthcare workers with the *distance* between healthcare workers u and v being the hop distance in the hospital graph between u

and v's respective home locations. We then partition this metric space in a rather simple way, by connecting any pair of healthcare workers that are distance at most s from each other, for some integer parameter $s \geq 0$. The connected components of this graph induce a partition of the healthcare workers and by considering all possible values of s we find one value ($s = 6$) that maximizes the modularity. Figure 2.13 shows that even this simple partitioning algorithm based on spatial attributes returns a community structure that is better than the attempts mentioned earlier.

Finally, we have implemented the greedy clustering algorithm, called `maxQ`, described by Clauset et al. [29] to determine if there are other vertex-partitions with even better modularity values. As shown in Figure 2.13, this algorithm yields a community structure whose modularity value is above the 0.3 threshold, noted as significant by Newman [80], with the modularity value in the $sparse_1$ graph being particularly high. This indicates that the HCW contact networks have a strong underlying community structure, with concomitant implications for infection control. For example, since communities are densely connected it makes sense not to try and break up communities, but rather to focus on breaking the links between communities.

2.3.3 Small World Property

The possibility that social networks may have unexpectedly small average path length or diameter was first highlighted by Milgram's well known "six degrees of separation" experiments [72]. Watts and Strogatz [109] show several real-world networks that have small average path length relative to Erdös-Renyi graphs of the same size and edge density and refer to this as the *small-world* property. Figure 2.11 shows that our HCW contact networks have one giant component along with lots of tiny connected components. For example, even though the $sparse_1$ HCW contact network in Figure 2.11 has 873 components, more than 84% of its vertices and more than 99% of its edges lie in its giant component. This is in contrast with the single connected component that the corresponding random graphs have. Both average path length and diameter of the giant components are very small relative to graph size (see the last two rows in Figure 2.11), clearly pointing to the "small world" nature of the HCW contact networks. For instance, the 5,838 individuals in the giant component of the $sparse_1$ graph are, on average, less than 4 hops from each other.

2.3.4 Diversity within Groups of Healthcare Workers

We partition healthcare workers into groups, each group defined by a unique (department, job title) pair. Table 2.1 shows the top 10 largest such groups and Table 2.2 points to a heavy-tailed distribution of the group sizes. Analysis of the degree distribution of the largest 10 groups (see Figure 2.14(a)) clearly indicates that degree distributions are heavy-tailed even within each group. Another key measure that we associate with each healthcare worker is their mobility. The *distance traveled* by person v in time window T is defined as follows. Suppose that person v has the following sequence of login sessions $(I_1, I_2, \ldots, I_t)$ in time window T. Recall that each login session I_j, $1 \leq j \leq t$, has a specific location $p(I_j)$, which is a vertex in the hospital graph. Let $D(x, y)$ denote the hop-distance in the hospital graph between vertices

Table 2.1: The largest ten groups of HCWs partitioned by distinct (department, job title) pairs.

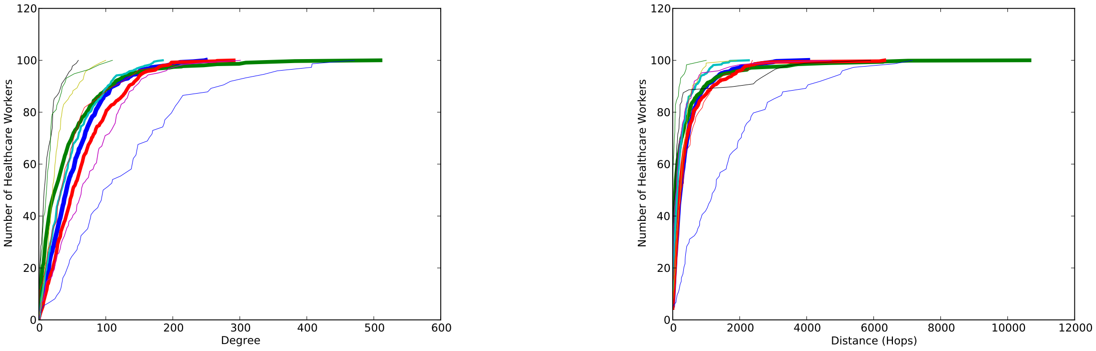
department	**job title**	**group size**
NURSING	STAFF NURSE II	795
N/A	N/A	664
NURSING	STAFF NURSE I	617
NURSING	NRS ASST	378
NURSING	NRS UNIT CLK	124
RESPIRATORY CARE	RESP THERAPIST	94
PATHOLOGY	CL LAB SCI II	88
INTERNAL MEDICINE	HSE STAFF FELL	74
NURSING	PSY NUR ASST I	58
INTERNAL MEDICINE	PROFESSOR	55

The second largest group is composed of HCWs that do not have an assigned "department" or "job title" in the EMR login data.

Table 2.2: The number of groups for sizes from 1 to 10.

Group Size	1	2	3	4	5	6	7	8	9	10
# of Groups	431	149	85	52	61	27	26	13	18	18

x and y. The *distance traveled* by person v is defined as $\sum_{j=1}^{t-1} D(p(I_j), p(I_{j+1}))$. Later, in Chapter 4, we use the distance traveled by a healthcare worker to determine which healthcare workers to vaccinate. The cumulative distance distributions (Figure 2.14(b)) for the largest 10 groups indicate that the distance distributions are even more heavy-tailed relative to degree distributions. One implication is that contact network modeling within a hospital setting that takes healthcare workers within a group (i.e., same ward or unit or job title) as being homogeneous is likely to miss important structural features defined by unique individual properties.



(a)

(b)

Figure 2.14: Cumulative density plots for the (a) degree distribution and (b) distance distribution for each of the largest 10 groups partitioned by (department, job title) pairs. The plots show, for a particular x-value, the fraction of healthcare workers whose degree (respectively, distance traveled) is at most the x-value. Thickness of the line indicates the size of the group.

2.3.5 Discussion

Based on this statistical description of the contact graphs, it is clear that the login contact graphs have all of the properties of "real world" networks highlighted by Watts and Strogatz [109], i.e., *despite being sparse, the login contact graphs have large clustering coefficient and have a giant connected component with small average path length.* As we will see in the following chapter, this has ominous implication for hospital-acquired infections: if these contact networks are a reasonable proxy for actual contacts between health-care workers, then diseases (at least those that respect the simplified model of infectious disease that Watts and Strogatz use) can spread quite rapidly in hospitals. On the positive side, other properties of these networks, such as heavy tailed degree distribution and strong community structure also suggest that targeted infection control strategies might be successful.

CHAPTER 3
RANDOM GRAPH MODELS OF CONTACT NETWORKS

Random graph models are a way of describing a family of graphs by focusing on a particular set of "important" graph characteristics and abstracting away and randomizing "unimportant" details. The determination of what details are "important" depends on the application of the graph being studied. Among many, there are two important reasons to want to design random graph models. The first is that with the enormous growth in networks being studied – compare the karate club graph of Zachary [111] in 1977 with 34 vertices and 78 edges to the web graph studied by Broder et al. [19] in 2000 with over 200 million vertices and over 1.5 billion edges – it is impossible to study networks visually and we require some framework for understanding the essential features of the graph topology. The second is that using random graph models allows for the development of analytical tools and algorithms that can be applied to families of networks, not just specific datasets. The graph models we discuss in this chapter are all "generative" models; they can be viewed as algorithms which generated a representative graph for that model.

One of the simplest and most well-studied graph models is the Erdös-Rényi (ER) random graph model, denoted by $G_{n,p}$ that considers graphs with n vertices where each pair of vertices is connected independently by an edge with probability p [35]. An interesting feature of ER random graphs is that a number of NP-hard optimization problems have been shown to have elegant solutions on ER random graphs. Even though Erdös and Rényi introduced their graph model because of its interesting mathematical properties, ER random graphs have often been taken to be "typical" instances of real-world graphs.

In 1998, Watts and Strogatz showed that real (world) networks exhibit the "small-world" property of having a characteristically short mean path length and are also highly clustered [109]. These properties make these real-world networks much different from ER random graphs. Along with their empirical analysis of a number of real networks, Watt and Strogatz provided one of the first models for generating "small world" graphs. Graphs based on the Watts-Strogatz model exhibit high clustering and short average path length while having a Poisson degree distribution similar to graphs described by the ER random graph model. As shown in the previous chapter, this Poisson degree distribution excludes this graph model as a candidate for our HCW contact networks, which have a heavy-tailed distribution. Fortunately, the work of Watts and Strogatz has sparked an explosion of graph models in the last century.

Barabási and Albert [12] give a model for "scale-free" networks (i.e., graphs that tend to follow a power-law degree distribution [10]) that are generated by a process commonly known as "preferential attachment". Power-law describes a degree distribution where the probability of a vertex having degree k is $p(k) = k^{-\alpha}$ for some $\alpha > 0$. In real-world scale-free networks, α tends to fall between 2 and 3 [10]. The Barabási-Albert model describes graphs which are grown by the following process; starting with b vertices, vertices are progressively added to the graph and attached to b vertices already existing in the graph, with probability proportional to the existing degree of each vertex (i.e., new vertices are more likely to attach to vertices with high degree).

More recently Chung and Lu introduce a model for generating random graphs with an expected degree distribution [27]. Given a degree distribution, the Chung-Lu (CL) model generates, with high probability, graphs with that degree distribution. In their work, Chung and Lu show that for graphs with a given degree distribution a giant component exists if the expected average degree is at least 1 and no component exists if the second-order average degree is at most 1.

The Configuration graph model, first introduced by Bender and Canfield [14] and made famous by Molloy and Reed [75], generates a graph uniformly at random from the collection of all graphs with the given degree sequence. The main contribution of Molloy and Reed was to show that for the degree sequence given by the vector $\{d_1, \ldots, d_n\}$, there is a giant component with high probability iff $\sum_i i(i-1)\frac{d_i}{\sum_i d_i} > 0$. This summation is commonly referred to as the "phase transition" where the giant component is formed. The giant component is a subgraph containing a majority of the graph's edges where every pair of vertices is connected by a path.

Newman et al. [89] derive an alternate proof of the conclusions of Molloy and Reed which they extend to calculate the average component size, giant component size, and average path length. This work has been extended further to include calculations of the expected outbreak size of diseases for graphs generated by the Configuration model [84, 68].

The Configuration model has been extended to generate graphs with a specified "assortative mixing" of vertices in the network [82, 88, 85]. The *assortativity* of a network, loosely defined, measures the degree to which endpoints of edges have similar properties. A specific instance of assortative mixing is *degree assortativity* which depends on the degrees of vertices. Roughly speaking, degree assortativity is the likelihood that high-degree vertices are, or are not, connected to other high degree vertices. When we refer to *assortativity* in this work we are referring to degree assortativity.

Most recently there has been interest in random graph models that incorporate a specified level of clustering [8, 87, 90]. Here we define clustering loosely as a local property of how well the neighbors of a vertex are connected (i.e., the degree to which neighbors are also neighbors).

There is also a more sophisticated model considered by Leskovec et al. [64] that uses *Kronecker products* to iteratively grow graphs from smaller subgraphs. Since their primary focus is on the evolution of graphs, this model focuses on graphs that result from "doubling" a smaller graph in a prescribed way to generate graphs that, besides having a heavy-tailed degree distribution and small diameter, exhibit *densification* and a shrinking diameter over time. These graphs in practice have a high computational cost and do not preserve local structural properties, like degree distribution [99].

3.1 Chapter Overview

In this chapter we look at a number of these candidate random graph models and investigate their usefulness and applicability in the study of disease diffusion. We narrow our focus to popular graph models that are (i) easy to generate and (ii) are mathematically tractable. Our motivation for generating and comparing graph models is to find a one to act as a proxy for the instances of HCW contact networks we generate from EMR login records. Having a such a model gives us insight into (i) the structural properties of the HCW contact networks relevant to the diffusion of disease, (ii) a model for developing better approximations for optimization problems on these networks, and (iii) the ability to generate new contact networks without reliance on copious amount of data.

First, in Section 3.2, we describe a number of popular graph metrics for comparing graphs. In Section 3.3 we introduce a number of graph models as candidate models for our HCW contact networks. We focus on graph models that have been successfully used by researcher for modeling social networks that arise in other contexts. In Section 3.4 we use output from a simple agent-based simulation, which mimics the spread of influenza, as a metric for comparing the progression of disease diffusion across networks generated by these random graph models. In Section 3.5 we consider the next natural evolution of these models, random graphs that generate graphs with a specified level of clustering, and show the difficulties in generating such graphs. And finally in Section 3.6 we describe a new graph model for generating graphs with specified clustering and consider it as a candidate model for our HCW contact networks.

3.2 Graph Metrics

There are a number of measurements that we use to compare random graph models with HCW contact networks.

Mean Degree ($\langle k \rangle$): The mean degree over all vertices in the graph. We consider the standard deviation of the mean (σ) as well as the maximum degree k_{max}.

Degree Distribution: The degree distribution is the probability distribution over degrees of all vertices. Since many social networks, like our HCW contact networks, exhibit a heavy-tailed distribution [5, 39, 47, 60, 61] of vertex degrees, the average degree is not indicative of the majority of vertices. Thus we also compare the degree distributions of these graphs.

Clustering Coefficient (cc): The Clustering Coefficient measures to what extent the neighbors of a vertex are also neighbors. This is another measure that Watts and Strogatz identified to differentiate real-world networks from ER random graphs. The clustering coefficient of a vertex v is the ratio of the number of edges between neighbors of v over the total possible edges that can exist between neighbors of v. The maximum possible edges between neighbors of v is $\frac{d(v)(d(v)-1)}{2}$. If we let $tri(v)$ denote the number of actual edges that exist between neighbors of v, then the clustering coefficient for vertex v is

$$cc(v) = \frac{2 \cdot tri(v)}{d(v)(d(v)-1)}.$$

The clustering coefficient of graph $G = (V, E)$ is defined as the average $cc(v)$

for all $v \in V$.

Transitivity (t): Transitivity is the ratio of closed triads with the total number of possible triangles. Similar to the clustering coefficient, let $tri(v)$ denote the number of edges that exist between neighbors of vertex $v \in V$. Let $triad(G)$ denote the pairs of edges that share a vertex. The *transitivity* is then

$$\frac{\sum_v tri(v)}{triad(G)}.$$

The transitivity is similar to the clustering coefficient, but in cases where triangles are distributed throughout the graph as opposed to localized to a few vertices, the transitivity can differ from clustering coefficient by order of magnitude [8, 102].

Assortativity (r): Assortativity [85, 82, 22] measures the extent to which vertices with similar properties or types share an edge. For a particular type or property, we say that a graph is assortative if vertices tend towards being connect with other vertices of the same type. In our work we focus on *degree assortativity* which measures the correlation between the degrees of endpoints of edges. Higher assortativity means that higher degree vertices tend to be connected to other higher degree vertices and, similarly, lower degree vertices tend to be connected to other lower degree vertices. The work on assortativity focuses primarily on the *excess degree* of a vertex which is one less than the degree of the vertex. In the standard definition given by Newman [82, 85] the degree assortativity measures the correlation between *excess degrees* of the vertices. For ease of exposition we refer to the degrees of the vertices directly, i.e., if a vertex has excess degree k we say the vertex has degree $k+1$. As defined by Newman [85], let e_{jk} denote the fraction of edges that connect vertices of degree $j+1$ and $k+1$. To remain consistent with the notation introduced by Newman [85], we suppose that each undirected edge has a starting vertex A and ending vertex B. Thus, each edge starts with a vertex A of type j and ends at a vertex B of type k, and only occurs in one entry of the matrix e. The matrix e satisfies the following equalities,

$$\sum_{jk} e_{jk} = 1 \qquad \sum_{k} e_{jk} = a_j \qquad \sum_{j} e_{jk} = b_k.$$

Here a_j is the fraction of edges that start at a vertex with degree $j+1$ and b_k is the fraction of edges that end at a vertex with degree $k+1$. The measure of assortativity proposed by Newman is the Pearson correlation coefficient over e_{jk},

$$r = \frac{\sum_{jk} jk(e_{jk} - a_j b_k)}{\sigma_a \sigma_b}$$

where σ_a and σ_b are the standard deviations over the distribution of a_j and b_k. Values for r range from -1, perfect disassortativity, to 1, perfect assortativity. When there is no assortative mixing, $e_{jk} = a_j b_k$ and $r = 0$. ER random graphs are an example of a graph having $r = 0$ (i.e., no assortative mixing).

Number of Connected Components (c): The number of connected components in the graph.

Singletons: Number of components made up of only one vertex (i.e., isolated vertices).

Size of the Largest Component (n_{giant}): By measuring the size of the largest connected component we check for the existence of a giant component which contains a majority of the vertices. The existence of the giant connected component also plays a key role in the spread of disease. If we assume that disease cannot spread outside of the connected component where it starts, then the size of the giant component is an upper bound on the number of people infected given an outbreak. Further, a large giant component with short average path length and small diameter imply that disease can spread quite rapidly to a large part of the population represented by the network.

Density of the Largest Component (m_{giant}): The number of edges in the largest component describes how densely connected the giant component is if it exists.

Mean Path Length ($\langle \ell \rangle$): The mean shortest path length between all pairs of vertices measured in the largest component by hop distance. Short average path length in real world networks was first highlighted during Milgrams [72] "small-world" experiments and is the property that characterizes small-world networks as defined by Watts and Strogatz [109].

Diameter: The Diameter measures the worst-case distance between any pair of vertices. More precisely, this is defined the maximum shortest-path between any pair of vertices in the largest component. Similar to the average path length measure, the impact of a small diameter is dependent on the existence of a giant component.

3.3 Generating Random Graphs

In this work we focus on the Erdös-Rényi, Barabási-Albert, Chung-Lu, Configuration, and Configuration with Assortativity models. The ER model is presented as a reference differentiating the more recent models from "random" graphs of yore. The other models are presented as a representative set of recent models for describing social networks.

While the HCW contact networks we generate have edge weights, to our knowledge there are no random graph models that incorporate weighted edges. Of course, it is easy to assign edge weights to edges from some distribution, but such an approach ignores correlations that might exist between edge weights and other graph features such as degree distribution, clustering coefficient, etc. It is unclear how these edge weights may affect our ability to understand the structural differences of these models. We try to first understand how well these unweighted models match our HCW contact networks purely on structure, before conflating them with the added complexity of edge weights. Thus, for the remainder of this chapter we ignore edge weights.

Every graph model we study has an input set of parameters which we derive from our HCW contact networks. In all cases we assume we are generating graphs from our HCW contact network $G = (V, E)$ with vertex set V and edgeset E such that $|V| = n$ and $|E| = m$.

Erdös-Rényi (ER): The Erdös-Rényi (ER) random graphs are generated based on parameters n and p. Starting with a graph of n vertices and no edges, each pair of vertices is connected by an edge with probability p. To ensure the expected density of the ER random graph is the same as the density of our HCW contact networks we set $p = \frac{2 \cdot m}{n \cdot (n-1)}$, the mean degree of our contact networks.

Barabási-Albert (BA): The Barabási-Albert model takes parameters n and b. Starting with b vertices, the graph is "grown" by iteratively adding $n - b$ new vertices. When a new vertex is added to the graph, it is connected to b vertices already in the graph. The probability of connecting to vertex u already in the graph with probability proportional to

$$p_u = \frac{d(u)}{\sum_v d(v)}.$$

where $d(u)$ denotes the current degree of vertex u. To ensure the expected density of BA random graphs have the same average degree as our HCW contact networks we choose $b = \frac{m}{n}$.

Chung-Lu (CL): The Chung-Lu model for generating graphs takes as input the degree sequence over n vertices $\{d_1, \ldots, d_n\}$ where d_k denotes the degree of vertex k. To generate graphs based on this model we start with a graph with n vertices and no edges. Edges are placed independently at random between vertices u and v with probability, $\frac{d_u d_v}{\sum_1^n d_i}$. This generates a graph expected degree sequence $\{d_1, \ldots, d_n\}$. More specifically, after placing edges, vertex i has degree d_i with high probability.

Configuration (CONF): The configuration model is given as input an explicit degree sequence over n vertices $\{d_1, \ldots, d_n\}$ and generates a graph uniformly at random from all graphs with given degree sequence. Generation starts with n vertices with a vertex v having d_v "open" stubs, representing candidate endpoints for edges. A pair of stubs is picked uniformly at random and an edge is placed between those stubs, effectively closing the two stubs, until there are no more open stubs. Since each vertex v has stubs equal to its degree d_v, it is guaranteed that each vertex v will have no more than degree v. From a theoretical perspective, as n tends towards infinity, we do not have to worry about self-loops or multiple edges [83]. In practice, graphs generated by this method may contain self-loops and multi-edges. Since our HCW contact networks contain neither, we simply ignore any graphs generated with multiple edges or self loops and simply regenerate the graph.

Configuration with Assortativity (CA): The CA model takes as input the degree sequence of n vertices $\{d_1, d_2, \ldots, d_n\}$ as well as an assortativity matrix e and generates a graph uniformly at random from all graphs with the given degree sequence and assortativity matrix. Denote the number of edges in the graph we are generating as $m = \frac{\sum_i d_i}{2}$. Each entry e_{jk} gives the fraction of edges that connect vertices of degree $j+1$ to vertices of degree $k+1$. Recall discussion in Section 3.2 for details on handling of matrix e in undirected graphs. Further, we let $a_j = \sum_k e_{jk}$ and $b_k = \sum_j e_{jk}$ be the fraction of edges that start with vertex of degree $j+1$ and end with vertex of degree $k+1$ respectively. Like in the Configuration model, we have n vertices with vertex v having d_v "open" stubs representing candidate edges and let S_j be set of open stubs adjacent to vertices of degree j. For our purposes we are generating graphs with given degree sequence and assortativity matrix calculated based on HCW contact networks and thus assume that there are (i) no self loops and (ii) $m \cdot (a_i + b_i) = |S_{i+1}|$ (i.e., there are exactly enough stubs to accommodate the edges that start or end at a vertex of degree $i + 1$). For more general discussion about generating graphs with given degree sequence and assortativity refer to [83, 85]. Let U be the multiset of tuples containing $m \cdot e_{jk}$ instances of the tuple $\langle j, k \rangle$ for every j, k, corresponding to entries of the e_{jk} matrix. The graph generation process is as follows. First a tuple $\langle j, k \rangle$ is picked uniformly at random and removed from U. Stubs u and v are then picked uniformly at random and removed from the S_j and S_k respectively. Each time this happens an edge is placed between a vertex of degree i and degree j and the stubs are effectively closed. This process can fall into situations where a multiedge or self loop is formed. To avoid this in practice the stubs are redrawn up to a designated number of times at which point we accept the choice and move on, marking the edge as added but in need of fixing. When the graph has been generated, we resolve any self-loops or multi-edges by performing an edge swap. Specifically, suppose that edge (u, v) with endpoint types j and k has a multiedge or self-loop. To resolve this we find another edge of type j and k, say (u', v'), such that $u \neq u', u \neq v', v \neq v'v \neq u$ and there are no edges (u, v') or (u', v). Then we perform a swap by removing edges (u, v) and (u', v') and adding edges (u, v') and (u', v). Since our input parameters are based on HCW contact networks, we know that these swaps are

Table 3.1: Graph statistics for the *sparse*$_{50}$ graph and graphs generated using the Erdös-Rényi (ER), Barabási-Albert (BA), Chung-Lu (CL), Configuration (CONF), and Configuration with Assortativity (CA) models.

	HCW	ER	BA	CL	CONF	CA
n	7144	7144.0	7144.0	7144.0	7144.0	7144.0
m	70505	70581.0 (±279.775)	64215.0	70547.167 (±219.872)	70505.0	70505.0
$\langle k \rangle$	19.738	19.76 (±0.078)	17.977	19.75 (±0.062)	19.738	19.738
σ	26.199	4.447 (±0.036)	20.781 (±0.306)	26.579 (±0.117)	26.199	26.199
k_{max}	232	38.65 (±2.372)	450.75 (±49.015)	236.222 (±13.99)	232.0	232.0
r	0.165	−0.001 (±0.003)	−0.02 (±0.003)	−0.002 (±0.004)	−0.005 (±0.003)	0.165 (±5.551 − 17)
cc	0.311	0.003	0.013	0.016 (±0.001)	0.016	0.012
t	0.25	0.003	0.011	0.021	0.02	0.028
c	1207	1.0	1.0	1406.889 (±12.684)	1129.9 (±0.831)	1177.55 (±0.589)
singletons	1128	0.0	0.0	1405.444 (±12.82)	1128.0	1128.0
n_{giant}	5758	7144.0	7144.0	5737.667 (±12.574)	6014.2 (±1.661)	5918.2 (±1.939)
m_{giant}	70173	70581.0 (±279.775)	64215.0	70546.722 (±219.722)	70504.1 (±0.831)	70455.75 (±1.374)
$\langle \ell \rangle$	3.718	3.282 (±0.005)	3.058 (±0.005)	3.064 (±0.005)	3.123 (±0.001)	3.263
diameter	13	5.0	5.0	6.824 (±0.381)	7.1 (±0.436)	9.95 (±0.865)

For random graph models values are averaged over 20 generated graphs and ± values indicate significant ($> .0001$) standard deviation.

always possible.

3.3.1 Comparison of Graph Properties

Tables 3.1 and 3.2 compare the graph properties for our HCW contact networks against the Erdös-Rényi, Barabási-Albert, Chung-Lu, Configuration, and Configuration with Assortativity models.

The first thing we note is that both configuration and the CA models generate graphs with exactly the same mean degree since these models generate the

Table 3.2: Graph statistics for the *moderate*$_{50}$ graph and graphs generated using the Erdös-Rényi (ER), Barabási-Albert (BA), Chung-Lu (CL), Configuration (CONF), and Configuration with Assortativity (CA) models.

	HCW	ER	BA	CL	CONF	CA
n	7144	7144.0	7144.0	7144.0	7144.0	7144.0
m	191270	191331.4 (±393.333)	185068.0	191189.684 (±445.964)	191270.0	191270.0
$\langle k \rangle$	53.547	53.564 (±0.11)	51.811	53.525 (±0.125)	53.547	53.547
σ	64.704	7.301 (±0.054)	50.943 (±0.301)	65.046 (±0.136)	64.704	64.704
k_{max}	660	82.2 (±1.806)	729.85 (±37.115)	664.474 (±21.975)	660.0	660.0
r	0.139	-0.0001 (±0.002)	-0.005 (±0.002)	-0.002 (±0.002)	-0.016 (±0.001)	0.139
cc	0.413	0.007	0.028	0.041 (±0.001)	0.041 (±0.001)	0.03
t	0.234	0.007	0.027	0.045	0.043	0.062
c	323	1.0	1.0	438.0 (±6.258)	300.2 (±0.4)	315.167 (±0.373)
singletons	299	0.0	0.0	436.947 (±6.287)	299.0	299.0
n_{giant}	6770	7144.0	7144.0	6706.947 (±6.236)	6844.55 (±0.921)	6814.5 (±1.118)
m_{giant}	191116	191331.4 (±393.333)	185068.0	191189.632 (±445.969)	191269.75 (±0.536)	191254.667 (±0.745)
$\langle \ell \rangle$	3.134	2.657 (±0.001)	2.57 (±0.001)	2.65 (±0.002)	2.666 (±0.001)	2.828 (±0.001)
diameter	11	3.95 (±0.218)	4.0	5.95 (±0.218)	5.85 (±0.357)	8.333 (±0.471)

For random graph models values are averaged over 20 generated graphs and ± values indicate significant (> .0001) standard deviation.

exact degree sequence as the input HCW contact network. The Chung Lu allows for more freedom by generating graphs with *expected* degree sequence, but still generates graphs with approximately the same mean degree. The Barabási-Albert model has a lower mean degree and lower standard deviation over this mean, but has a maximum degree almost double that of the HCW contact network. To show more detail about the degrees of graphs generated by these graphs, Figures 3.1 and 3.2 compare the degree distributions of the HCW contact network and ER, BA, and CL models. The configuration-based models are not plotted as they match the degree distribution of the HCW contact network exactly. The BA model has a minimum degree of $b = \frac{m}{n}$, which is an effect of the model itself, since each new vertex connects to exactly b vertices in the graph.

One area where all generated graphs fail to accurately model the HCW contact networks is the clustering coefficient. It is especially interesting to note that even though the CA model preserves the assortativity of the graph, it has very little clustering coefficient. Similarly, all graphs generated by the model have an order of magnitude smaller transitivity.

An interesting characteristic of the HCW contact networks is their relatively high assortativity (r). Except for the CA mode which explicitly maintains assortativity, the remaining models fail to model this attribute and in fact generate graphs that are neither assortative nor disassortative. The minor disassortativity (negative assortativity) in the Barabási-Albert model is likely the effect of new vertices (with relatively low degree) having a higher probability of sharing an edge with a high de-

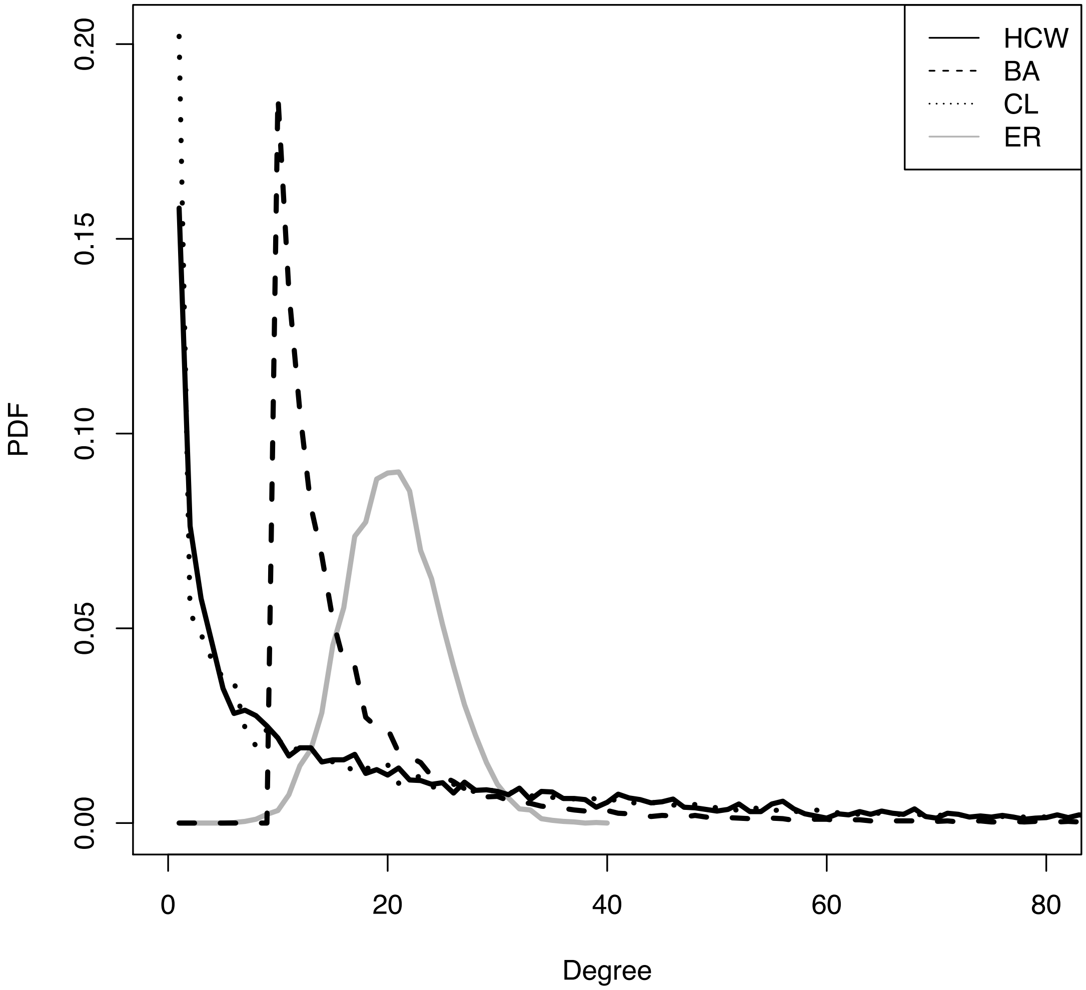


Figure 3.1: Degree distributions for the $sparse_{50}$ HCW contact network and corresponding graphs generated by the Erdös-Rényi (ER), Chung-Lu (CL), and Barabási-Albert (BA) models. x-axis denotes the degree. y-axis denotes the fraction of nodes. Plot is truncated; maximum values can be found in Table 3.1.

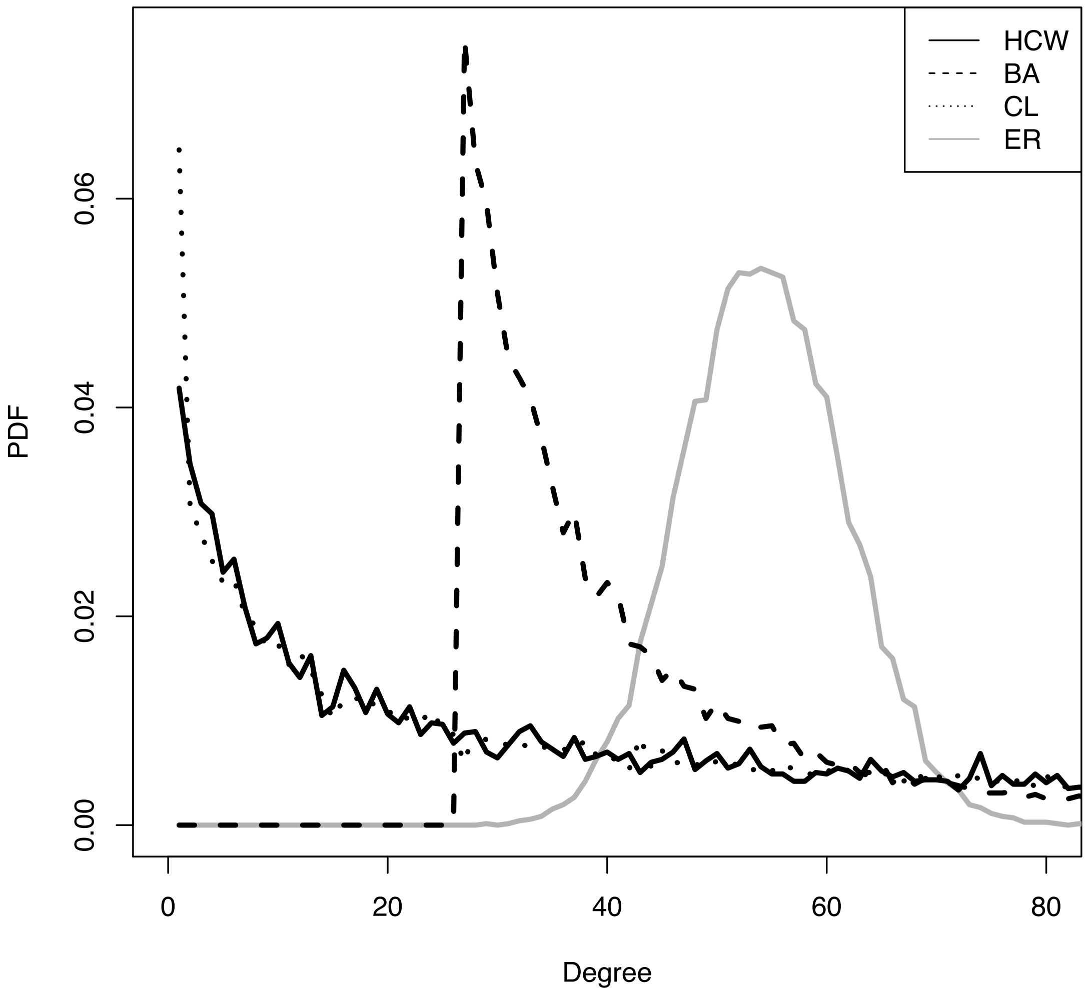


Figure 3.2: Degree distributions for the $moderate_{50}$ HCW contact network and corresponding graphs generated by the Erdös-Rényi (ER), Chung-Lu (CL), and Barabási-Albert (BA) models. x-axis denotes the degree. y-axis denotes the fraction of nodes. Plot is truncated; maximum values can be found in Table 3.2.

gree vertex. And despite the configuration model matching the HCW contact network degree sequence exactly, generated graphs have no assortativity.

There is also a noticeable difference between the giant component sizes of the various graph models. The ER and BA models generate a single giant component containing all vertices whereas the CL model generates a component of roughly the same size as the HCW contact networks. However, the CONF and CA models generate graphs with a giant component slightly larger than the HCW contact networks we generate (roughly 5% larger). But what is more interesting about this is that, despite having larger components, the random graph models have shorter mean path length $\langle \ell \rangle$. One hypothesis for the behavior is that the HCW contact networks have "tails" of vertices extending from the giant component that do not happen in the random graph models due to mixing of edge endpoints. The differences in diameters, with the HCW contact networks having diameter almost twice that of the CONF model, is further evidence of the existence of these "tails".

3.4 Comparing the Spread of Disease

While there are obvious structural differences between the HCW contact network and graphs generated by the random graph models, one of our primary motivations is to be able to use these random graphs models as the basis for improvements to optimization problems relating to disease diffusion. However, relatively little research has been done on how well random graph models model diffusion processes on the graph instances which they represent. Specifically, we would like to know how disease diffusion behaves on graphs generated by these random graph models in comparison with our HCW contact networks.

To simulate the spread of disease we use an agent-based simulation, based on a simple SIR-like model defined by parameter p, that attempts to model the spread of influenza. In our model, each individual is assumed to have the same susceptibility to disease, have the same transmissibility, remain sick the same amount of time, and stay active in the contact network for the entirety of the simulation. Transmissibility is assumed to last for exactly m days. On the ith day of being infected, $1 \leq i \leq m$, individual j spreads the disease to neighbor k with probability $p^i_{j,k}$. We set $m = 9$ and set $p^i_{j,k}$ values according to vector of shedding levels $S = (0.016645, 0.05, 0.035235, 0.02137, 0.011155, 0.007115, 0.005015, 0.003195, 0.00336)$ derived from plots in Carrat et al. [23]. Specifically, the ith entry in this vector, S_i denotes the shedding level on day i. Based on the parameter p, which we call the "peak transmission probability", the vector S is scaled so that $S_2 = p$. We compute the $p^i_{j,k}$ values using the formula $p^i_{j,k} = (1-(1-S_i)^{\frac{w(j,k)}{28}})$. Recall that $w(j,k)$ is the weight of edge $\{j,k\}$, corresponding to the total number of contacts between j and k during time period T and therefore $w(j,k)/28$ represents the average number of daily contacts between HCW j and k during a 4-week (28 day) period. Based on this model we rely on the following Lemma to calculate a number of properties on our graphs.

Lemma 3.1. *The transmission probability (transmissibility) for disease spread across edge (j,k) with weight $w(j,k)$ is $\rho = 1 - \prod_i (1-S_i)^{\frac{w(j,k)}{28}}$.*

Since the random graph models generate unweighted graphs, we use a uniform edge weight $w(j,k) = 28$ for all edges (j,k), representing a single contact per day. In the remainder of this chapter we refer to this agent-based simulation by the transmissibility ρ as calculated by Lemma 3.1.

Each simulation we run generates the number of new infections each day as a result of a single infected individual chosen uniformly at random from the population. Simulations start with a single infected individual and continue until there is no longer anyone infected. We run $10,000$ simulations for each graph and track the number of infected each day, recording both the median and average values for each day over all runs. Our plots give the number of infected individuals on a given day. In these experiments we don't record the specific daily values for each individual simulation. We refer to the mean or median "disease curve" as the sequence $I_1, I_2, \ldots, I_k$ where k is the number of days until disease dies out and I_i is the mean or median number of individuals infected on day i, respectively.

Table 3.3: Numerical comparison of the curves in Figure 3.3 measured by the sum of the squared difference in the number of individuals infected each day.

	HCW	CA	CL	CONF	ER	BA
HCW	0.0	405.5	847.705	883.252	4801.827	2486.387

Smaller values indicate more similarity.

Figures 3.3 and 3.4 show mean and median disease diffusion curves, respectively, for the $moderate_{50}$ HCW contact network and the random graph models based on simulations for $\rho = .0302$. The differences in the two figures shows an important

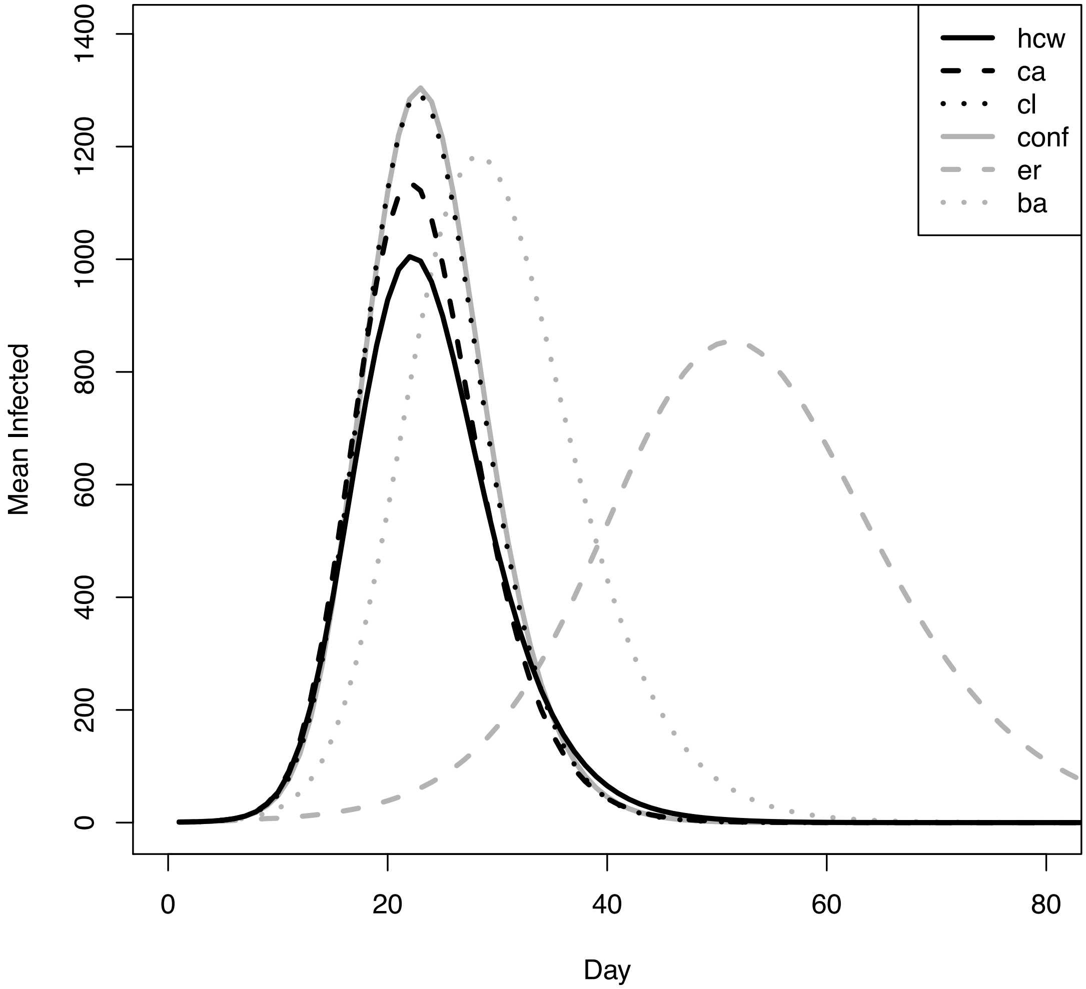


Figure 3.3: Comparison of graphs generated by the Erdös-Rényi (ER), Barabási-Albert (BA), Chung-Lu (CL), Configuration (CONF), and Configuration with Assortativity (CA) models with the $moderate_{50}$ (HCW) graph they model. Graph shows the mean number of people infected on a given day of the SIR simulation with transmissibility ($\rho = .0302$). x-axis represents the timestep for the simulation (days). y-axis gives the number of people in the "infected" state during that timestep.

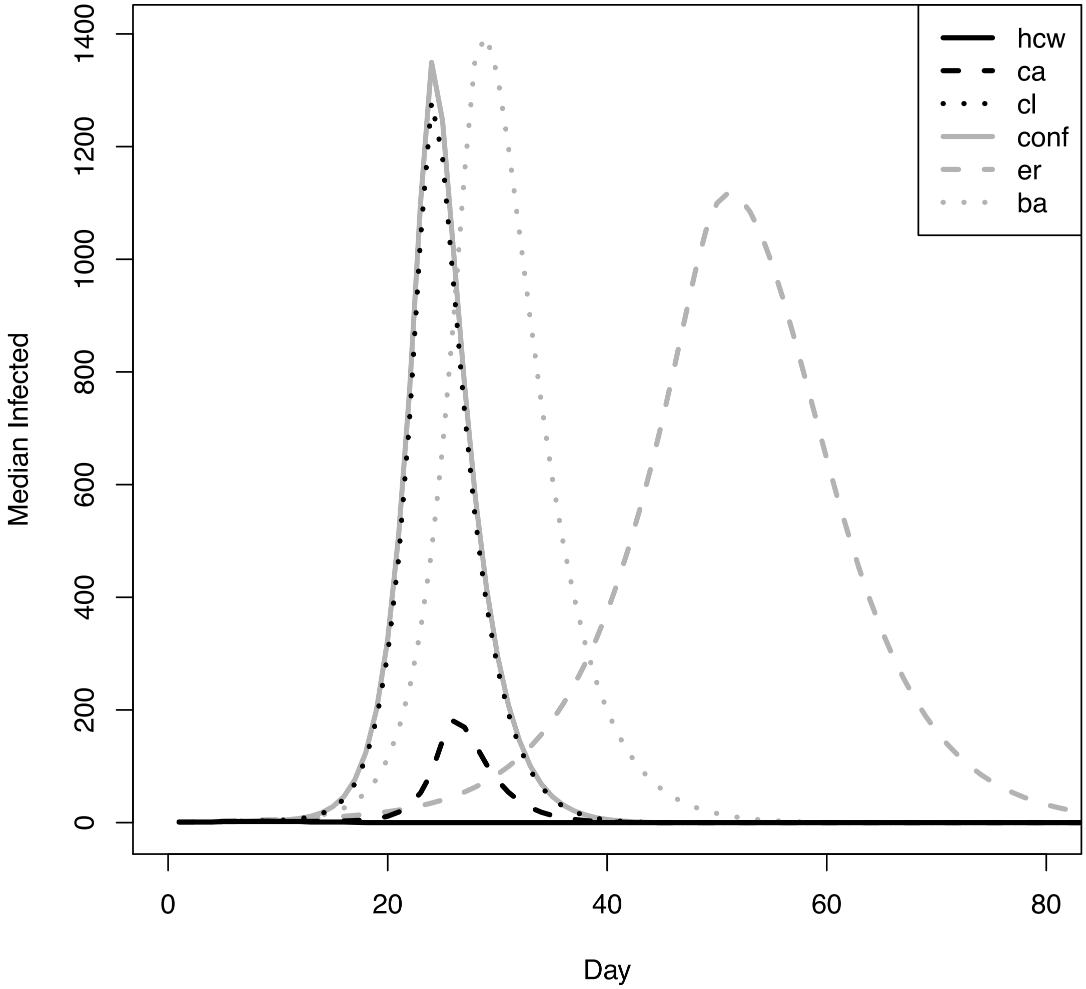


Figure 3.4: Comparison of graphs generated by the Erdös-Rényi (ER), Barabási-Albert (BA), Chung-Lu (CL), Configuration (CONF), and Configuration with Assortativity (CA) models with the *moderate*$_{50}$ (HCW) graph they model. Graph shows the median number of people infected on a given day of the SIR simulation with transmissibility ($\rho = .0302$). x-axis represents the timestep for the simulation (days). y-axis gives the number of people in the "infected" state during that timestep.

Table 3.4: Statistics for the plots in Figures 3.3 and 3.4 showing mean and median number of individuals infected, the mean positive increase in the curve and day with the most number of infected individuals.

	HCW	CA	CL	CONF	ER	BA
Mean Inf.	1697.46	1795.006	1994.875	2017.214	3039.618	2503.289
Median Inf.	3	3460	3719	3739	4587	4166
Mean $+\Delta$	5.997	8.145	9.311	8.002	1.595	5.663
Peak Day	22	22	23	23	51	28

difference between disease diffusion on ER random graphs and those graphs generated from the CONF model, CA model, and HCW contact networks. When using the *mean* number of new infected per day, the peak on CONF, CA, and HCW contact network is higher than that of the ER graphs. However, when using the *median* number of new infected each day the peak of the ER graph exceeds that of the CONF, CA and HCW contact network. This suggests that more than half the time, the number of new infected on a day for the CA and CONF graphs is quite low and 0 in the HCW contact networks. But the exceedingly high peak when using the mean number of new infected suggests that there are a few cases where there is a huge number of infected individuals, causing the high peak in the average case.

To measure the differences in the disease curves we take the squared difference at each point (see Table 3.3). For pairs of curves this measurement gives a metric for how closely the curves follow each other. As we can see, both visually in Figure 3.3 and in Table 3.3, the disease curve of the CA model most closely matches the curve of the HCW contact network followed by the CONF and CL models.

Most research on random graph models focuses on simply comparing the mean number of infected individuals as a result of a single infected individual chosen uniformly at random. Aggregate values for the simulations in Figure 3.3 are given in Table 3.4. Simply focusing on mean or median values ignores temporal aspects of disease diffusion that we see in our disease curves. Thus, we also include the mean positive slope value ($+\Delta$), which takes the average positive increase in each curves, and the day in which the disease curve peaks. This provides additional information that is important to characterizing the spread of disease on these models. For example, the ER graph model overestimates the mean and median number of infected, but disease also spreads much more slowly on the ER graph.

Table 3.5: Statistics for the plots in Figure 3.5 showing mean and median number of individuals infected, the mean positive increase in the curve and day with the most number of infected individuals.

	HCW	CA	CL	CONF	ER	BA
Mean Inf.	317.291	328.448	334.222	332.196	2.875	56.854
Median Inf.	1	1	1	1	1	1
Mean $+\Delta$	0.229	0.248	0.18	0.176	0.011	0.01
Peak Day	39	40	45	45	9	47

Table 3.6: Numerical comparison of the curves in Figure 3.5 measured by the sum of the squared distances each day.

	HCW	CA	CL	CONF	ER	BA
HCW	0.0	33.115	129.863	143.088	415.394	371.799

Smaller values indicate more similarity

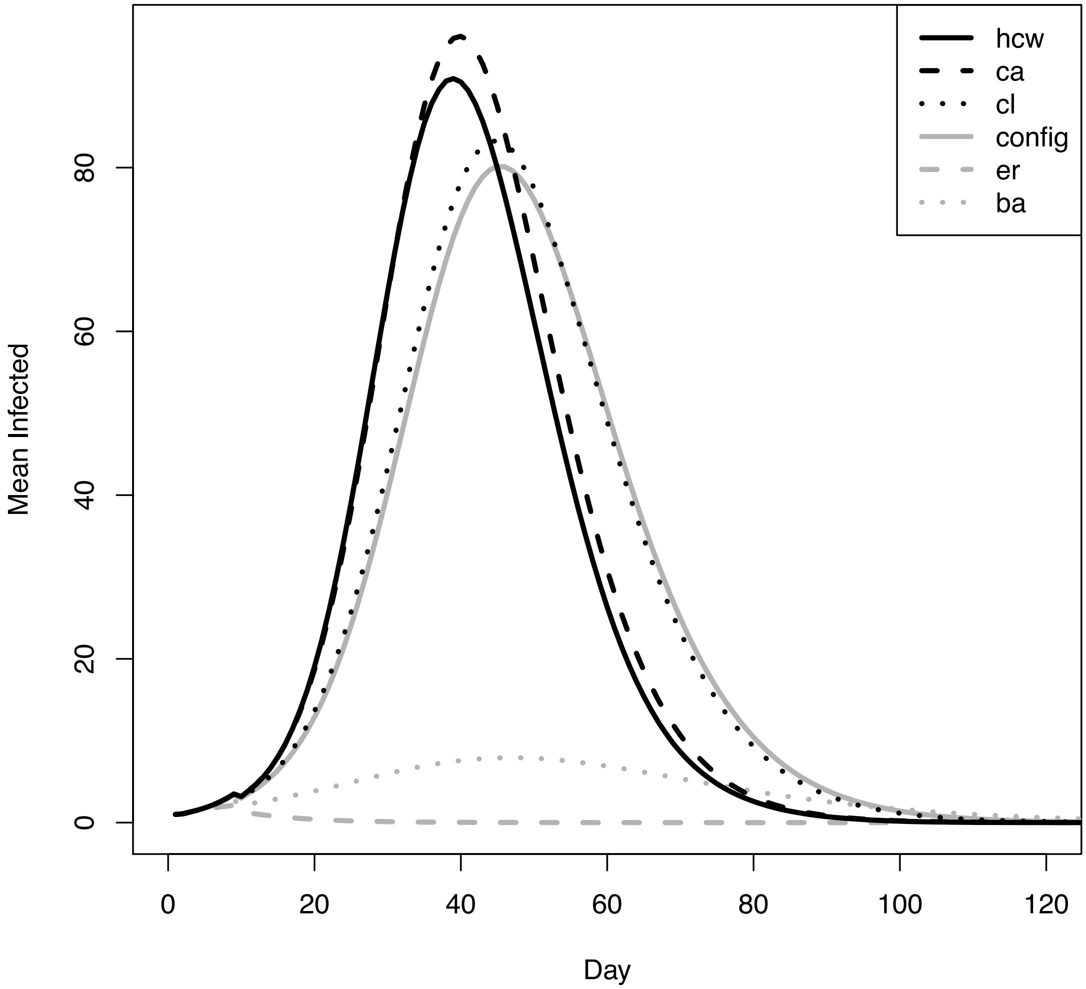


Figure 3.5: Comparison of graphs generated by the Erdös-Rényi (ER), Barabási-Albert (BA), Chung-Lu (CL), Configuration (CONF), and Configuration with Assortativity (CA) models with the $moderate_{50}$ (HCW) graph they model. Graph shows the mean number of people infected on a given day of the SIR simulation with transmissibility $\rho = .0122$. x-axis represents the timestep for the simulation (days). y-axis gives the number of people in the "infected" state during that timestep.

The importance of measuring the rate of disease diffusion and peak time becomes more apparent when we lower transmissibility. Figure 3.5 shows the disease curves for the $moderate_{50}$ graph and corresponding random graph models for an SIR simulation for $\rho = .0122$, which is much lower than the simulation in Figure 3.3. Aggregate statistics for the disease curves are shown in Table 3.5. While the mean number of infected for the CA model is close to the HCW contact network, the CL and CONF models are actually closer. However, if we consider the rate of disease spread ($+\Delta$) and peak day from Table 3.5 we can see that indeed the CA model does more accurately match the HCW for these metrics. We can also confirm this in Table 3.6 where the disease curve for the CA model more closely resembles the disease curve for the HCW contact networks.

Another interesting aspect of transmissibility $\rho = .0122$ is that it is below the "epidemic threshold" for the ER random graphs. The epidemic threshold is the transmissibility at which disease transitions from infecting very few individuals to infecting lots of individuals. While not all graph models exhibit an epidemic threshold, for a number of random graph models, including the ER, CONF, and CA models, researchers have analytically determined the expected epidemic threshold for graphs generated from these models [91]. For ER random graphs the epidemic threshold is well-known $\frac{1}{\langle k \rangle}$ where $\langle k \rangle$ is the average degree. Since the ER graphs shown in the previous figures is based on the $moderate_{50}$ graph with mean degree 53.564, the epidemic threshold is $\frac{1}{53.564} = .01866$, explaining the absence of significant disease spread on the ER graph.

Finally, at lower transmissibility the CONF and CL models underestimate disease spread of the HCW contact networks. The CONF and CL graphs also exhibit a "lag" in hitting their peak after the HCW contact network and graphs generated by the CA model. Recall in Figure 3.3 both CONF and CL models overestimate disease spread on the HCW contact networks but peaks on roughly the same day. This may be partially explained by epidemic threshold of the CONF graphs. As given by Meyers [69] the epidemic threshold for the CONF model is $\frac{\langle k \rangle}{\langle k^2 \rangle - \langle k \rangle}$ where $\langle k^2 \rangle$ is the mean squared degree. The configuration graphs we generate, based on the $moderate_{50}$ HCW contact network, have mean degree 53.564 and mean squared degree 7053.889 and thus an epidemic threshold of 0.0076. In Figure 3.6 we compare $moderate_{50}$ graph and corresponding CONF graphs for transmissibility values $\{.00763, 0.00794, 0.00824, 0.00854\}$. Corresponding mean number of infected values are given in Table 3.7. It appears that as transmissibility approaches the epidemic threshold for the CONF model, the configuration model underestimates the spread of disease to a larger degree.

Table 3.7: Mean number of infected individuals for the disease curves in Figure 3.6.

Transmissibility	HCW	CONF
0.00854	452.60954	208.31246
0.00824	343.11787	149.03785
0.00794	274.22523	101.66481
0.00763	196.29836	77.80498

It turns out that graphs generated by the CA model also underestimate the dis-

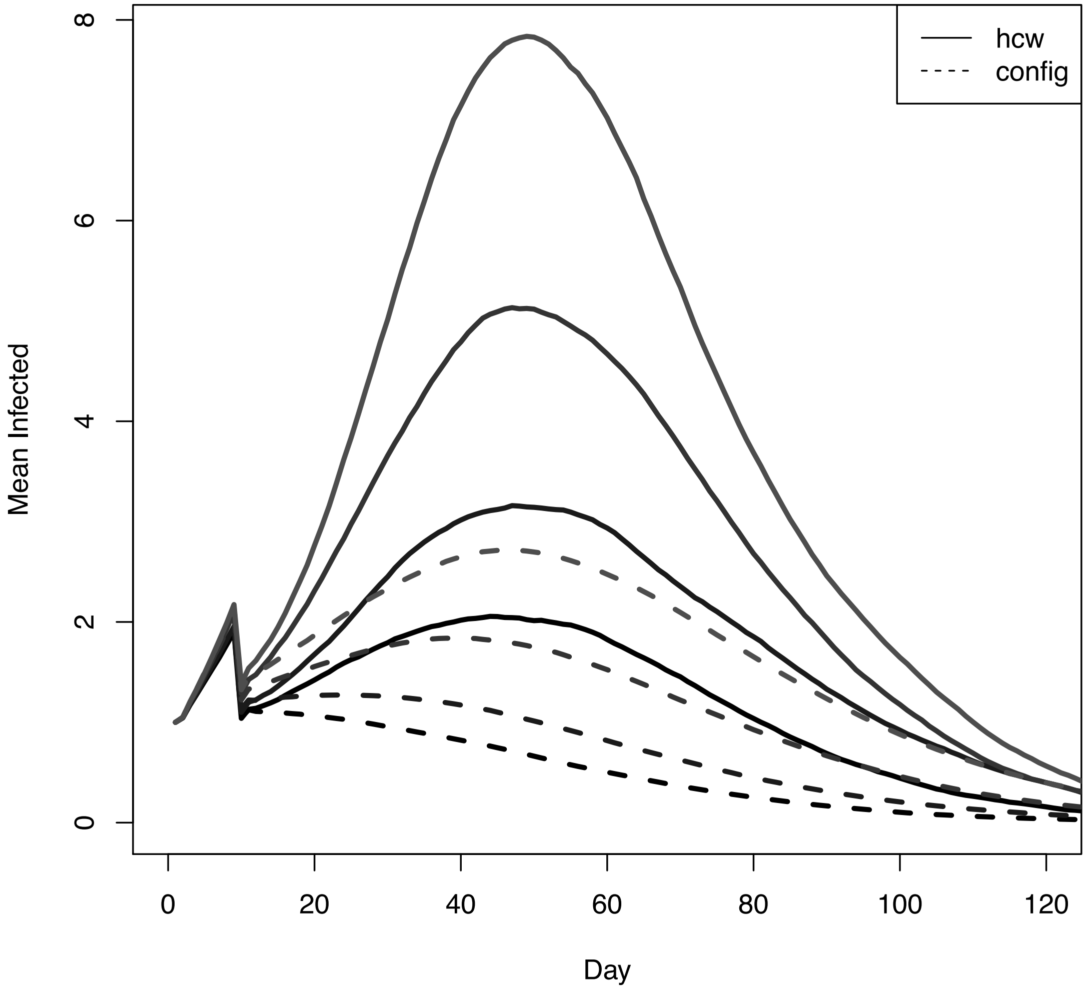


Figure 3.6: Comparison of the HCW contact network and CONF graphs for transmissibility values in $\{.00763, 0.00794, 0.00824, 0.00854\}$.

ease spread of the HCW contact networks, just to lesser extent. Figure 3.7 shows disease curves for the $moderate_{50}$ graph and corresponding graphs generated by the CA model based simulations for transmissibility values $\{.00763, 0.00794, 0.00824, 0.00854\}$. Corresponding mean number of infected values are given in Table 3.8.

Table 3.8: Mean number of infected individuals for the disease curves in Figure 3.7.

Transmissibility	HCW	CA
0.00854	452.6095	409.7092
0.00824	343.1179	324.0961
0.00794	274.2252	236.4079
0.00763	196.2984	167.4243

In general, random graph models tend to fix one property of the graphs they are generating and randomize over the others. Obviously as we control for more properties, such as degree sequence *and* assortativity in the CA model, there is less "mixing" of edges. Here we use the term mixing loosely to mean the randomization of connections within the network. We hypothesize that this mixing is what causes the CONF model to largely overestimate disease spread at higher transmissibility and underestimate disease spread at lower transmissibility. Where the CA model allows for less mixing since the correlations between the vertex degrees at endpoints of edges must be maintained.

Recall that assortativity measures the correlation between the degrees of endpoints of edges. High assortativity means that edges tend to connect vertices of similar degree, i.e., high degree vertices are more likely to connect to other high de-

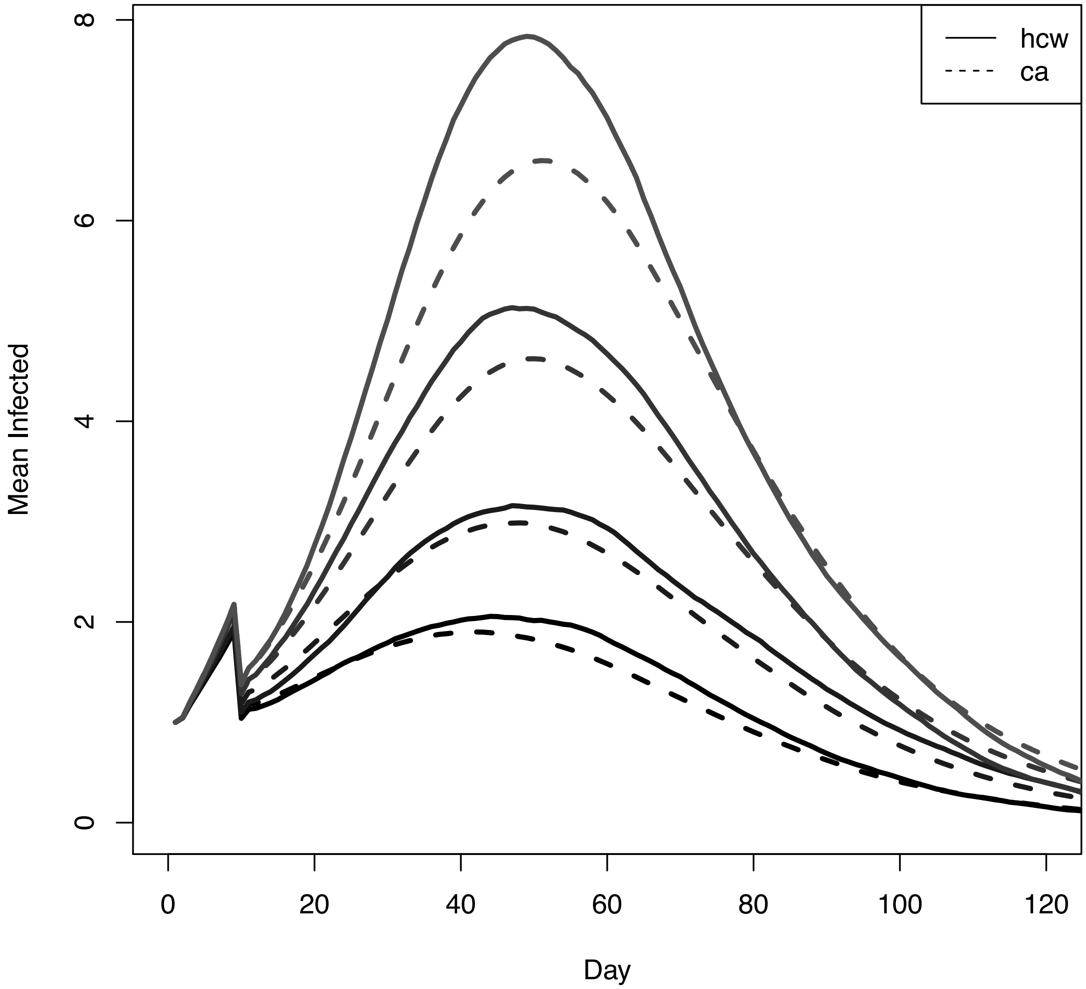


Figure 3.7: Comparison of the HCW contact network and CONF graphs for transmissibility values in $\{.00763, 0.00794, 0.00824, 0.00854\}$.

gree vertices. This supports the experiments performed by Newman where synthetic graphs with positive assortativity, no assortativity, and negative assortativity were compared for various edge densities [85]. From these experiments Newman concluded that positively assortative graphs exhibit a giant component at lower edge density than graphs with no assortativity or negative assortativity. Further, these graphs appear to have a dense "core group" of vertices which, for low transmission probability, act as a reservoir for disease, allowing the disease to remain active despite low transmissibility. At higher transmission probability, disease spread is more limited in assortative graphs likely due to smaller size of the giant component. Put another way, even with higher transmissibility, disease is less likely to spread outside this densely connected "core" in assortative graphs.

In general, these disease diffusion simulations show compelling evidence that assortativity is an important aspect of the HCW contact networks we generate. And while the CA model is a remarkably more accurate model for the HCW contact networks there are still some minor differences.

3.5 Clustering in Networks

To this point we have considered a number of random graph models that model features such as mean degree, degree distribution, and assortativity. The next natural feature to consider is clustering. A common metric for local clustering around a vertex, which we have defined precisely in Section 3.2 is the *clustering coefficient*. Here we let $cc(v)$ denote the clustering coefficient of vertex v

Analyzing and comparing the clustering coefficient of graphs is popular in social network research primarily due to the Watts and Strogatz [109] discovery that high clustering coefficient is one property that differentiates real-world networks from those generated by the Erdös-Rényi random graph model. Clustering coefficient alone doesn't give any information about the distribution of clustering among vertices or any correlation between clustering coefficient and degree. There are some interesting aspects to how the clustering coefficient is distributed among the vertices in the HCW contact networks. As we show in figure 3.8 the clustering coefficient for our HCW contact networks exhibits a Poisson-like distribution. However, there is evident correlation between clustering coefficient and degree in the HCW contact networks as shown in Figure 3.9. Recall that the clustering coefficient of a vertex v is roughly the number of pairs of neighbors of v that share an edge over the squared degree of v. Thus, a higher degree vertex requires many more edges between pairs of neighbors to achieve a high clustering coefficient. This may account for some of the decrease in clustering coefficient as degree increases.

Recently, a number of models for generating random graphs with given clustering coefficient have been proposed in the literature [8, 51, 87, 107, 106, 16, 90]. The proposed models can be classified into three categories which we call, *construction-based* [87, 90], *growing-based* [51, 107, 106, 16], and *rewiring-based* [8]. Construction-based models are given an input set of parameters, typically a degree distribution and clustering coefficient, start with a graph of n vertices and no edges, and then strategically place edges so that after edges have been placed, specified degree distri-

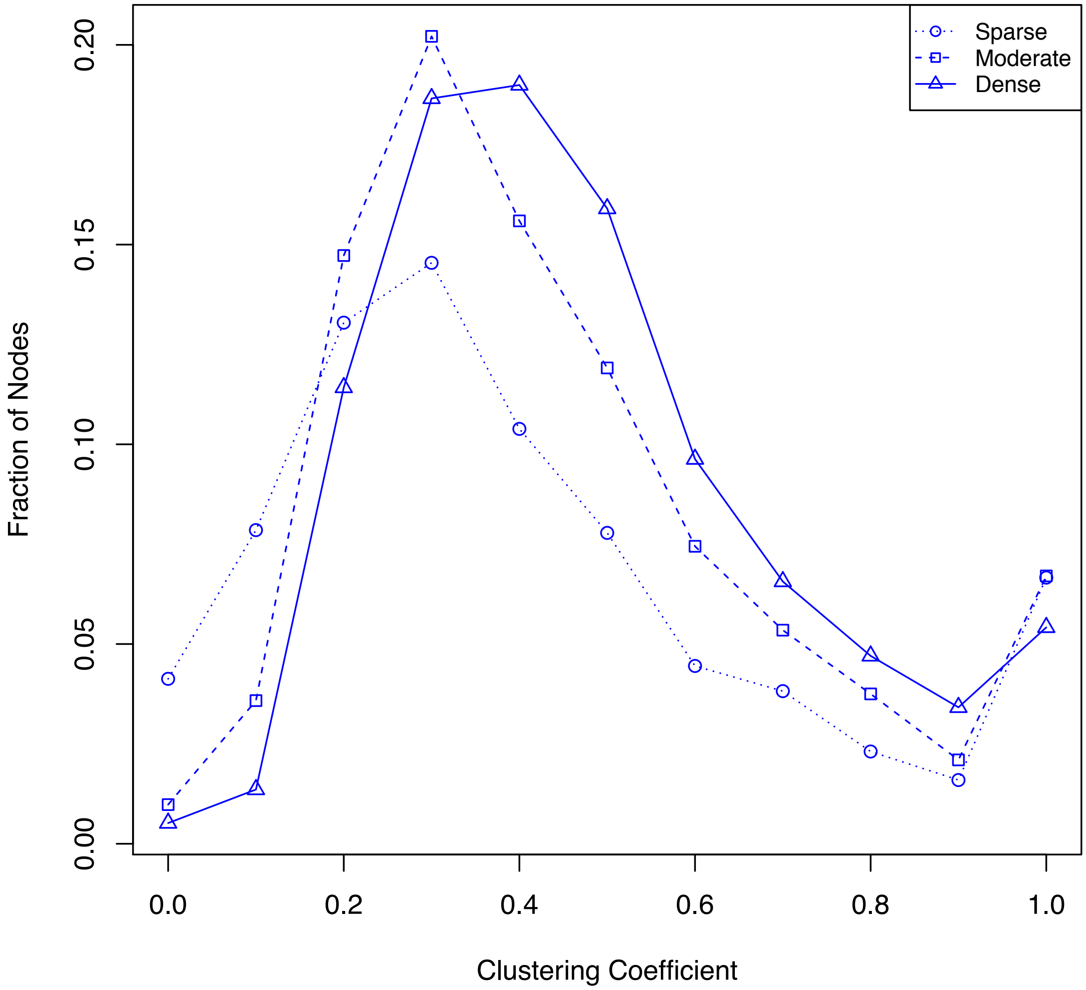


Figure 3.8: Clustering coefficient distribution for the $sparse_{50}$, $moderate_{50}$, and $dense_{50}$ HCW contact networks. Vertices with degree less than 2 are ignored.

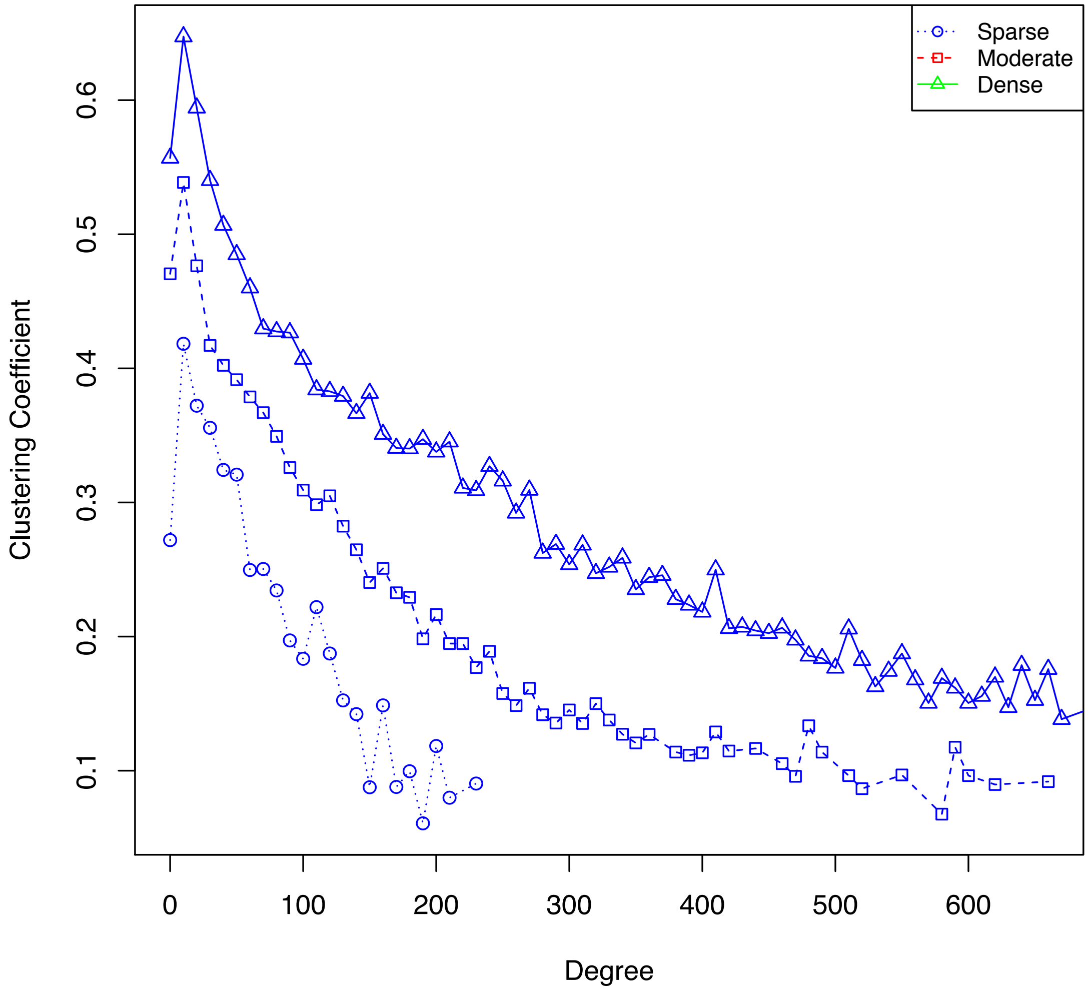


Figure 3.9: Vertex degree vs clustering coefficient correlation plot for the *sparse*$_{50}$, *moderate*$_{50}$, and *dense*$_{50}$ HCW contact networks. Vertex degrees are group in bins of size 10. The dense curve is truncated and extends out to 1240.

bution and clustering coefficient are met. Growing-based models build up a graph by repeatedly adding a vertex or small subgraph with high clustering which is attached to the rest of the graph based on input parameters to the model. For this work we ignore growing-based models because they are computationally expensive and don't preserve degree-sequence. Rewiring-based algorithms start by generating a graph using a model doesn't pay attention to clustering coefficient, and then increases the clustering coefficient by strategically "rewiring" edges [8].

3.5.1 Construction-Based Models

Because of the mathematical machinery available for them, we first considered construction-based random graph models which incorporate clustering. Newman [87] proposed a generalization of the configuration graph model, which we will call the *Configuration with Clustering (CC)*, for graphs with a given degree sequence $(d_1, \ldots, d_n)$ and "triangle" sequence $(t_1, \ldots, t_n)$. Here t_i is the number of triangles that have vertex i as a corner and d_i is the number of edges that do not participate in a triangle. Two example graphs with $d_u = 1$ and $t_u = 3$ are given in Figure 3.10(a) and 3.10(b).

One of the assumptions made by this model is that triangles in the graph do not share edges. Newman notes, "It is possible for single edges by chance to form triangles themselves, but it is straightforward to show that, so long as mean degree remains constant as n increases, the density of such triangles vanishes in the limit of large system size." Thus, the total degree of any vertex is $d_i + 2t_i$ and the graph in

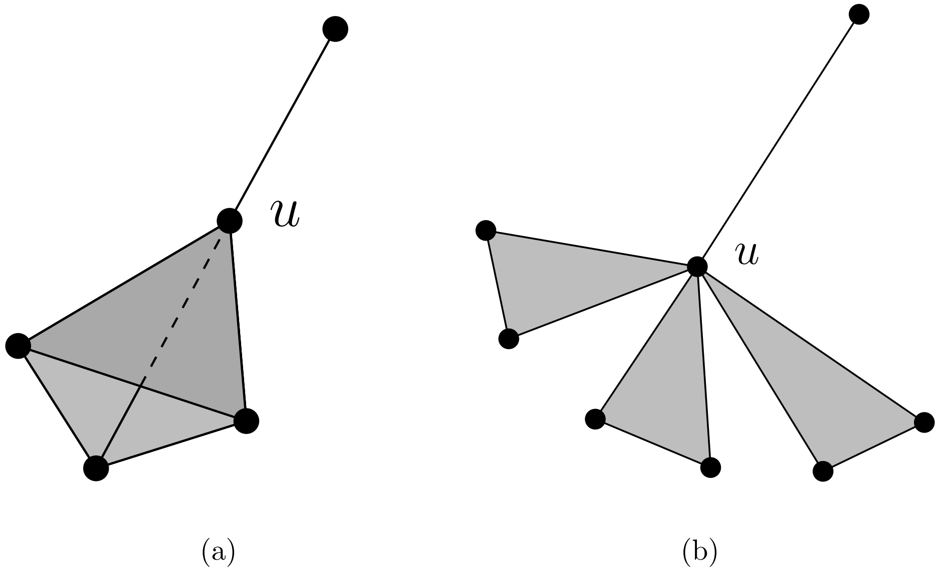


Figure 3.10: Two example graphs showing a joint degree-triangle distribution for a node u, where u has a single non-triangle forming edge $d_u = 1$, and is the corner of three triangles $t_u = 3$. (a) Configuration requiring 6 edges to 4 neighboring vertices and a clustering coefficient of .5. (b) Configuration requiring 10 edges to 7 neighboring vertices.

Figure 3.10(a) cannot be modeled. In fact, we can easily show that many real-world graphs cannot be modeled based on this assumption. Assume for the moment that a graph G has t triangles and no two triangles share an edge, then the minimum number of edges required to generate these triangles is $3t$. Table 3.9 shows the number of actual edges, the number of triangles, and the number of required edges under the assumption that no two triangles share an edge, for the karate club network from Zachary [111] and co-authorship network for network theory [86] along with the $sparse_{50}$ and $moderate_{50}$ HCW contact networks. From the table we can see that despite have a low number of edges, these graphs have a large number of triangles, suggesting that triangles share a large number of edges in these networks.

Table 3.9: Table comparing the number of triangles, actual edges, and minimum required triangle edges if no two triangles share an edge for two real-world networks and our HCW contact networks.

	Clustering	Triangles	Actual Edges	Req. Triangle Edges
Zachary Karate Club	.571	45	78	135
Co-authorship Network	.638	3,764	2,742	11,292
$sparse_{50}$	.311	313,903	70,505	941,709
$moderate_{50}$	.413	1,948,462	191,270	5,845,386

One way to avoid this assumption is to just ignore it. In an attempt to fix problems with the Newman approach, Parikh [90] proposed the **DEG** algorithm which takes as input a degree sequence $(d_1, d_2, \ldots, d_n)$ and a target clustering coefficient c, and aims to output a graph with the given degree sequence and clustering coefficient no less than c. The algorithm assumes that the clustering coefficient c is evenly distributed among the n vertices of the graph. For a vertex i to achieve a clustering coefficient of c it must have at least $c\binom{d_i}{2}$ triangles which it participates in, i.e., a c fraction of all pairs of neighbors must also be neighbors. Since each triangle is formed by three vertices, then the total number of triangles that must exist in a graph with the degree sequence given above is

$$T = \frac{c\sum_i \binom{d_i}{2}}{3}.$$

The **DEG** algorithm is then to place T triangles and then place edges appropriate to satisfy the desired degree sequence. To assure that degree sequence isn't violated the algorithm maintains a residual degree $rd[i]$ for each vertex i that is remaining edges which can be attached to i to assure that i has degree no larger than d_i. **DEG** is described in Algorithm 3.1. To place triangles the algorithm picks three vertices u, v, w with probability proportional to their residual degree $rd[u], rd[v], rd[w]$

Algorithm 3.1 DEG algorithm

```
input: Degree sequence $d = \{d_1, d_2, \ldots, d_n\}$ and value $c$, $0 < c < 1$
$V \leftarrow \{1, 2, \ldots, n\}$
$M \leftarrow \frac{\sum_i d_i}{2}$
$E \leftarrow \emptyset$
$rd \leftarrow d$
$T \leftarrow \frac{c \cdot \sum_i \binom{d_i}{2}}{3}$
while $T \geq 0$
    pick $u$ with probability $\frac{rd[u]}{\sum_x rd[x]}$
    pick $v$ with probability $\frac{rd[v]}{\sum_x rd[x]}$
    pick $w$ with probability $\frac{rd[w]}{\sum_x rd[x]}$
    if $rd[u], rd[v], rd[w]$ are sufficient for forming triangle $u, v, w$
        if $(u, v) \notin E$
            $rd[u] \leftarrow rd[u] - 1; rd[v] \leftarrow rd[v] - 1; M \leftarrow M - 1$
        end if
        if $(v, w) \notin E$
            $rd[v] \leftarrow rd[v] - 1; rd[w] \leftarrow rd[w] - 1; M \leftarrow M - 1$
        end if
        if $(u, w) \notin E$
            $rd[u] \leftarrow rd[u] - 1; rd[w] \leftarrow rd[w] - 1; M \leftarrow M - 1$
        end if
        $E \leftarrow E \cup \{(u, v), (v, w), (u, w)\}$
        $T \leftarrow T - 1$
    end if
end while
 while$M \geq 0$
    pick $u$ with probability $\frac{rd[u]}{\sum_x rd[x]}$
    pick $v$ with probability $\frac{rd[v]}{\sum_x rd[x]}$
    if $(u, v) \notin E$
        $E \leftarrow E \cup \{(u, v)\}$
        $rd[u] \leftarrow rd[u] - 1$
        $rd[v] \leftarrow rd[v] - 1$
        $M \leftarrow M - 1$
    end if
end while
return $(V, E)$
```

respectively. In Step 11 the algorithm checks that u, v, w have sufficient residual degree so that adding a triangle doesn't violate the degree sequence. Consider the case when edge $(v, w) \in E$ but $(u, v) \notin E$ and $(u, w) \notin E$ and $rd[u] = 1$. In this case, forming a triangle between vertices u, v, w requires that 2 edges adjacent to u, specifically (u, v) and (u, w), be added to E, but since the residual degree of u is only 1, this would violate the specified degree of u. Also, for this case, since $(v, w) \in E$, the formation of triangle u, v, w does not need to add edge (v, w) and thus Steps $12 - 20$ assure that the residual degrees $rd[i]$ of each vertex i and the number of unplaced edges M are maintained properly. Note that if a triangle already exists between u, v, w then the placement of the triangle is still counted against the number of triangles added.

We implemented this algorithm in an attempt to study the effect of clustering on disease diffusion. In our experiments, the number of triangles generated were orders of magnitude less than what was expected, as shown in Table 3.10. As a result, graphs generated by this algorithm failed to achieve the desired clustering coefficient. For the $moderate_{50}$ graph, the **DEG** algorithm calculates that it will need to place roughly 50% more triangles than are actually in the target graph, in order to achieve the desired clustering coefficient. The same phenomenon happens with the $sparse_{50}$ graph but not to the same degree. More importantly, even with these high expectations for placing triangles, the algorithm fails to generate enough triangles.

Figure 3.11 compares the clustering coefficient distribution of the $sparse_{50}$ to

Table 3.10: Results from experimental runs of the **DEG** algorithm.

	Input *cc*	Output *cc*	HCW Triangles	Calculated Triangles	Output Triangles
$sparse_{50}$	0.311239	0.15521	$313,903$	$391,433$	$48,478$
$moderate_{50}$	0.413189	0.135176	$1,948,462$	$3,443,959$	$421,103$

Shows the input clustering coefficient, the clustering coefficient of the output graph, the number of triangles in the corresponding HCW contact network, the number of triangles calculated by the algorithm based on the input clustering coefficient, and the number of triangles in the output graph

the corresponding graphs generated by the DEG algorithm. The plot shows that while there are a number of vertices with high clustering coefficient, a majority have a very low clustering coefficient, and there are relatively few vertices with clustering coefficient between .2 and .6 compared to the HCW contact networks. A more interesting view of this is given in Figure 3.12 where we consider the degree-clustering coefficient correlation for the $sparse_{50}$ graph and the corresponding graphs generated by the DEG algorithm. From this we see that the clustering coefficient that does exist in these graphs is the contribution of the a number of low degree vertices with high clustering coefficient. Put another way, the DEG algorithm fails to raise the clustering coefficient of higher degree vertices. Remember that to achieve the same clustering coefficient for a high degree vertex i, compared to a low degree vertex, requires a large number of edges between neighbors of i. One hypothesis is that in picking two other vertices to form a triangle adjacent to i, the DEG algorithm is unlikely to pick vertices that are already neighbors of i. Thus, each time a new triangle is formed which i participates in, it is unlikely to be formed between vertices that are already neighbors of i and very quickly the residual degree of i is exhausted before adequate clustering coefficient is achieved.

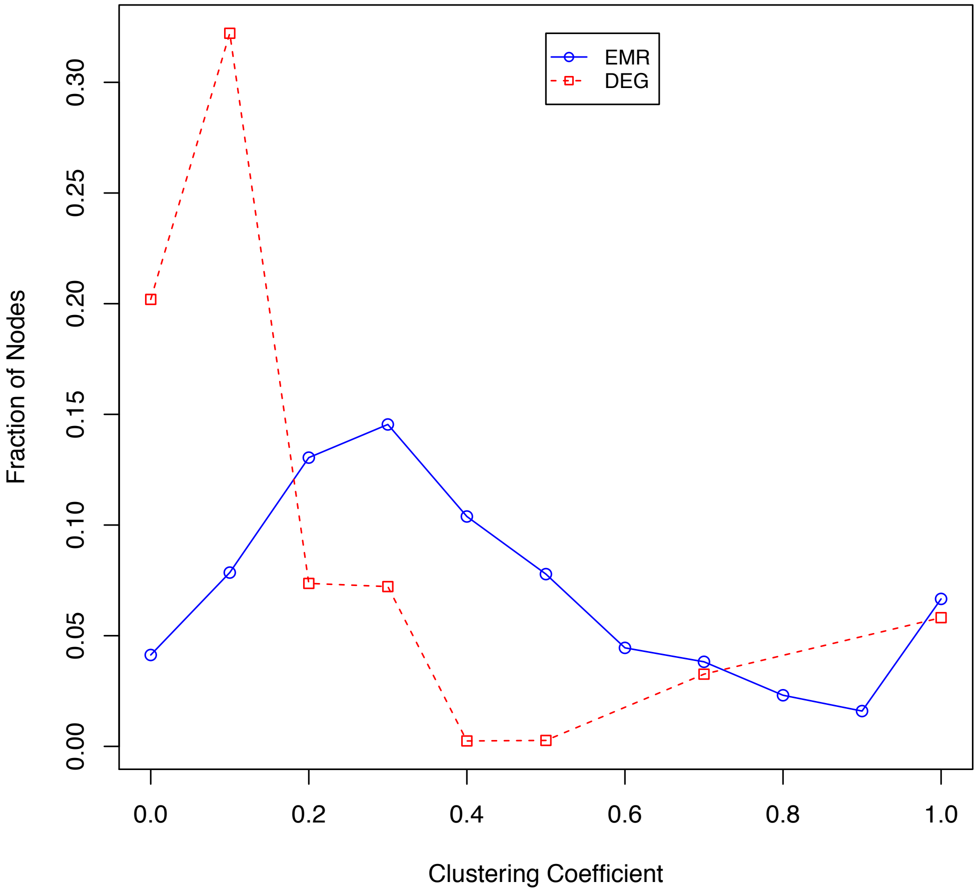


Figure 3.11: Clustering coefficient distribution for $sparse_{50}$ graph and corresponding graphs generated by the DEG algorithm. Vertices with degree less than 2 are ignored.

Graphs generated by the DEG algorithm, based on the $sparse_{50}$ graph do achieve a 0.15521 clustering coefficient. However, figures 3.13 and 3.14 suggest that

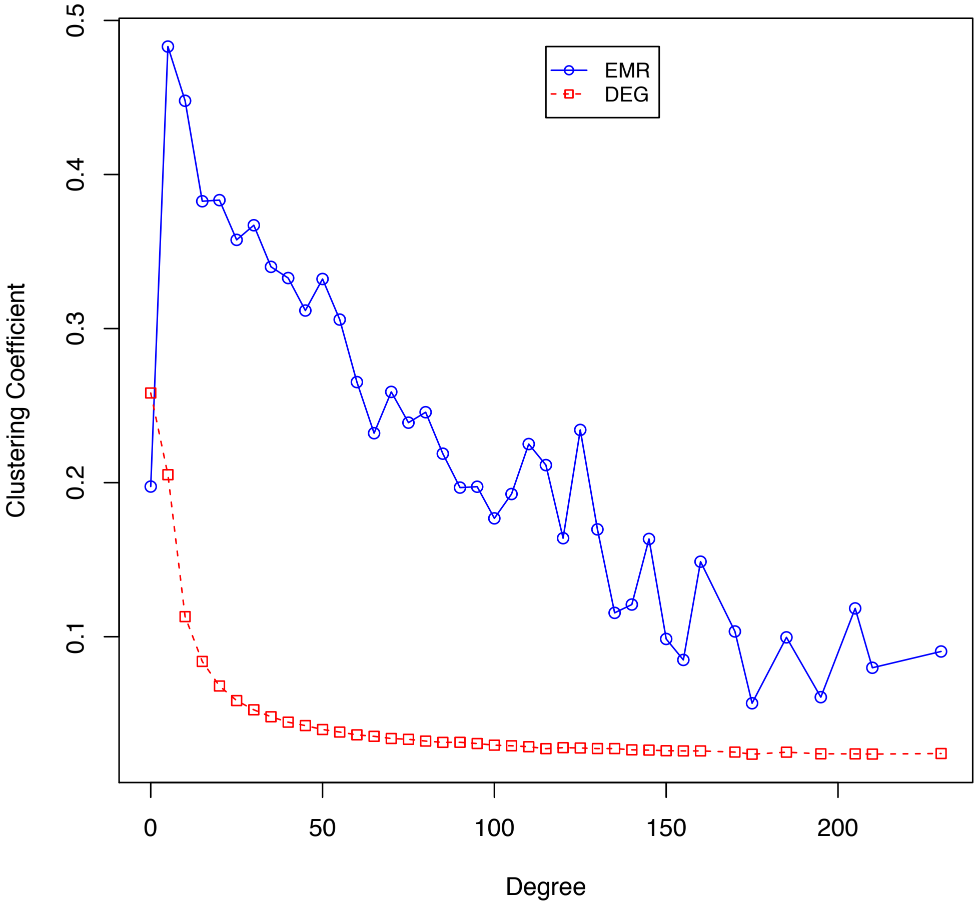


Figure 3.12: Vertex degree vs clustering coefficient correlation plot for $sparse_{50}$ graph and corresponding graphs generated by the DEG algorithm. Vertices are grouped into bins based on degree by groups of 5.

the higher 0.15521 clustering coefficient of the DEG graphs, compared to the 0.0157 clustering coefficient in the CONF graphs, results in negligible change to the disease curves. When transmissibility is low, as in Figure 3.13, minor differences, relative to the CONF graphs, may be due to this minor increase in clustering, allowing disease to spread to a few more vertices in the average. At higher transmissibility, as in Figure 3.14, this local clusters may act a “trap” for disease from which disease has a hard time escaping, and thus we see a very minor decrease in the disease curve peak for DEG graphs.

3.5.2 Edge Swapping

In practice, the construction-based approaches of the previous section are unable to generate graphs with clustering coefficient equivalent to our HCW, and thus we considered an edge swapping approach. Bansal, Khandelwal, and Meyers [8] propose a method for creating random graphs with a given degree sequence and clustering coefficient by running a Markov chain simulation algorithm that performs a series of edge swaps. The process and conditions for edge-swapping are given below and illustrated in Figure 3.15. In the configuration in Figure 3.15 , edges (y_1, z_1) and (y_2, z_2) can be remove and edges (y_1, y_2) and (z_1, z_2) added without perturbing the degrees of vertices y_1, y_2, z_1, z_2.

The method of proposed by Bansal, Khandelwal, and Meyers, which we will refer to as the BKM model, takes as input a degree sequence $S = (d_1, \ldots, d_n)$ and desired clustering coefficient c, and outputs a graph that has degree sequence S and

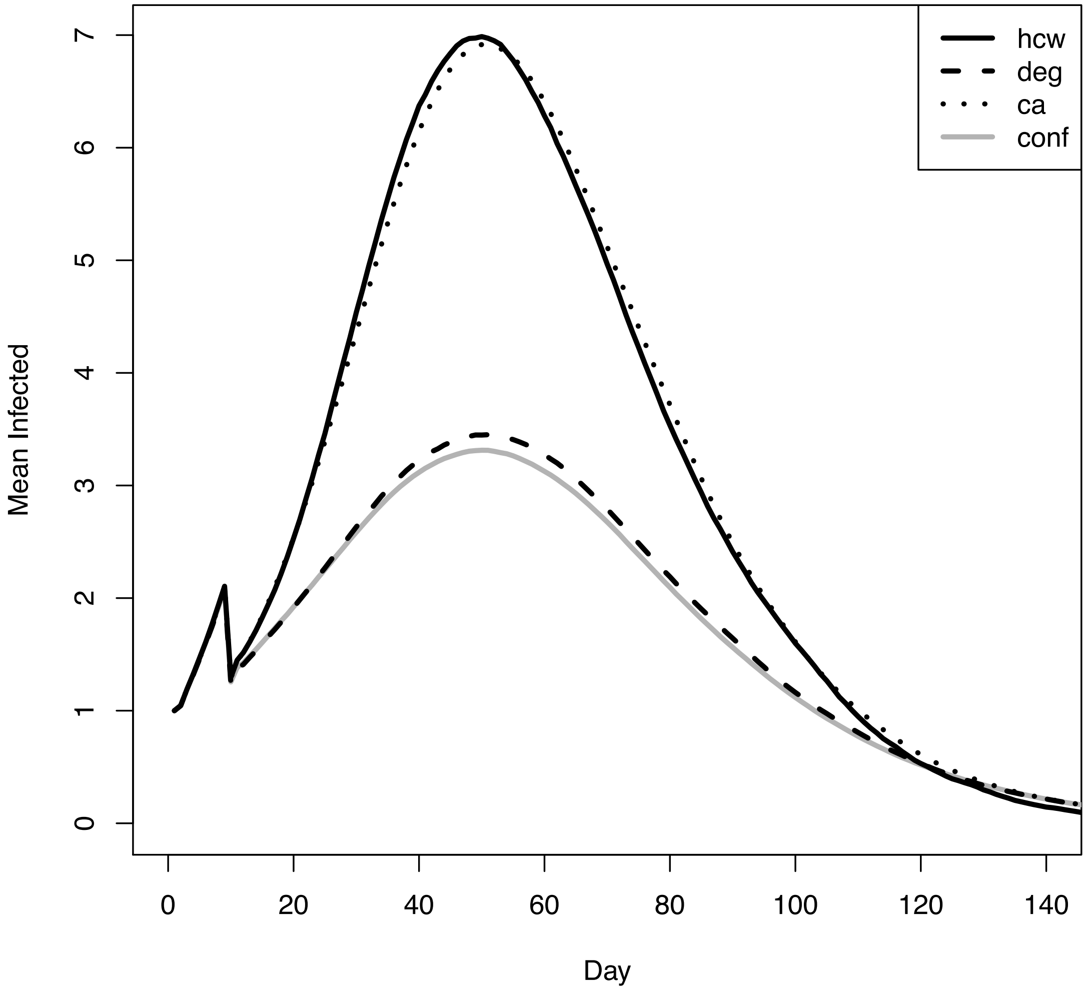


Figure 3.13: Comparison of $sparse_{50}$ graph and corresponding graphs generated by the CONF and CA random graph models and the DEG algorithm. Graph shows the average number of people infected on a given day of the SIR simulation with probability of transmission $\rho = .0122$. x-axis represents the timestep for the simulation (days). y-axis gives the number of people in the "infected" state during that timestep.

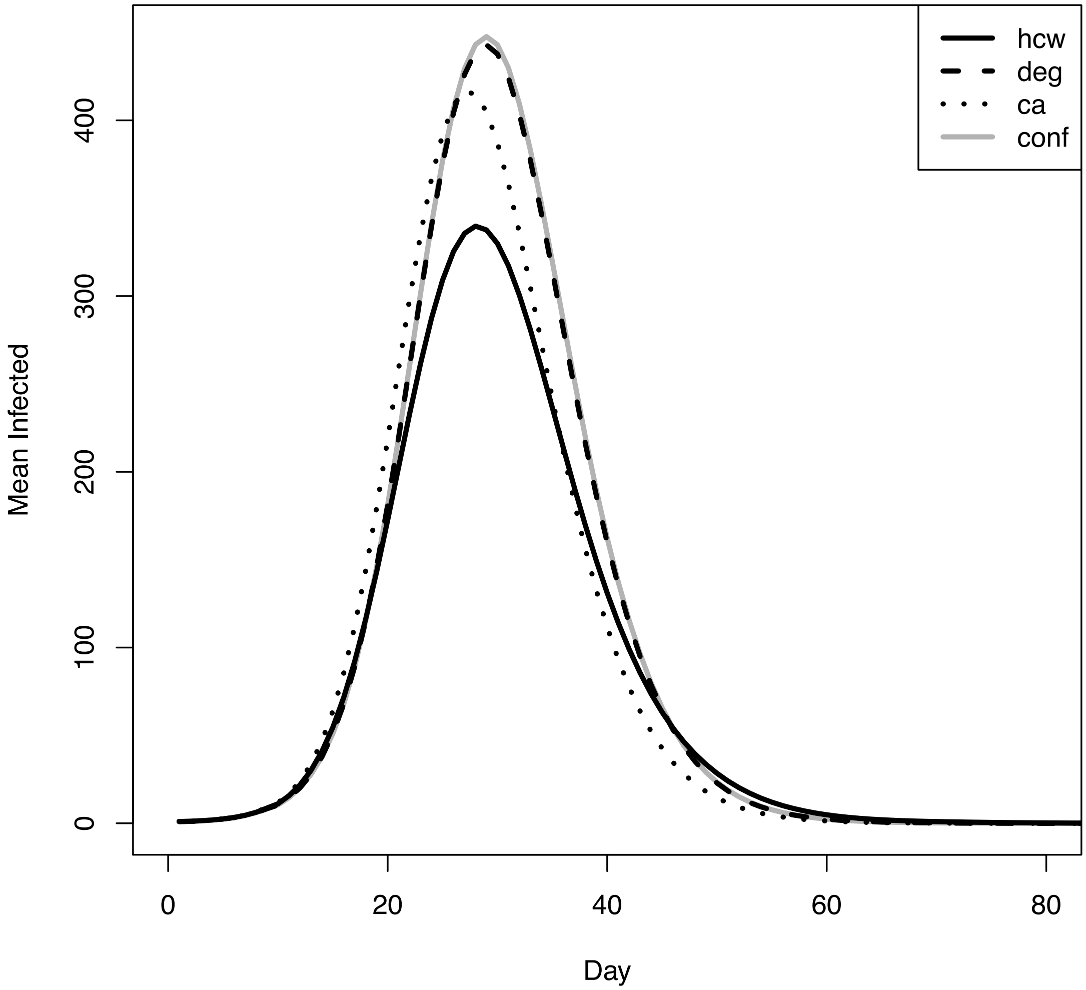


Figure 3.14: Comparison of $sparse_{50}$ graph and corresponding graphs generated by the CONF and CA random graph models and the DEG algorithm. Graph shows the average number of people infected on a given day of the SIR simulation with probability of transmission $\rho = .048$. x-axis represents the timestep for the simulation (days). y-axis gives the number of people in the "infected" state during that timestep.

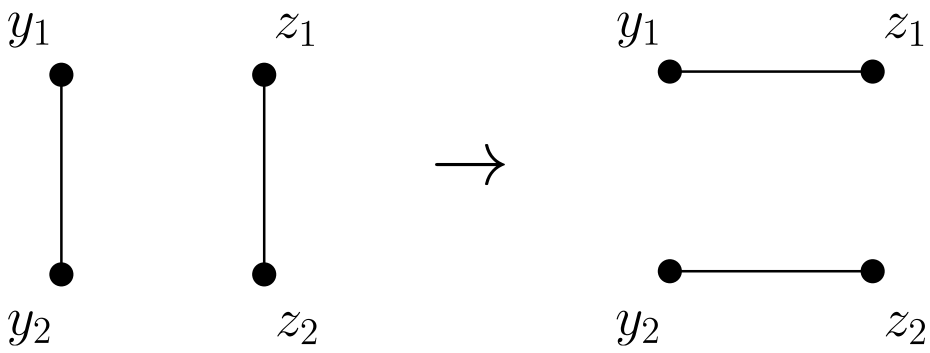


Figure 3.15: Edgeswap operation on vertices y_1, y_2, z_1, z_2.

clustering coefficient no smaller than c but is random in every other way [8]. The model starts with an initialization step which generates a graph $G = (V, E)$ based on the configuration model. Recall that the configuration model takes a degree sequence S and generates a random graph with exactly the degree sequence S. A Markov chain process is then carried out where an edgeswap is performed on a pair of randomly chosen edges (y_1, z_1) and (y_2, z_2) iff this edgeswap increases the clustering coefficient. This process continues until the desired clustering coefficient is reached.

The method for randomly picking edges (y_1, y_2) and (z_1, z_2) to swap starts by first choosing a random vertex x uniformly at random from the set of vertices with degree greater than 1. Then a pair of neighbors (y_1, y_2) of x are chosen uniformly at random from the set of all pairs of neighbors which are not connected by an edge. Finally, two vertices z_1, z_2 are chosen uniformly at random from $N(y_1)$ and $N(y_2)$, respectively, such that $z_1 \neq x$, $z_2 \neq x$, and $z_1 \neq z_2$.

In our experiments, implementations of this method are quite slow on large graphs due to the large number of vertices that influence the clustering coefficient.

One explanation is that in doing an edgeswap, not only are triangles formed by adding edges (y_1, z_1) and (y_2, z_2), but triangles that may be contributing to the clustering coefficient of other vertices are being destroyed by removing edges (y_1, y_2) and (z_1, z_2). More precisely, let $com(a, c)$ be all vertex b such that the edges $(a, b), (b, c) \in E$, i.e., all vertices with edges to both a and c. Now consider in the BKM process, when removing edge (y_1, y_2), each vertex $v \in com(y_1, y_2)$ has its clustering coefficient drop by $\frac{2}{d(v)(d(v)-1)}$. Since performing an edgeswap is removing edges (y_1, y_2) and (z_1, z_2), The total decrease in clustering coefficient is,

$$\frac{1}{|G|}\left[\sum_{v \in com(y_1,z_1)} \frac{2}{d(v)(d(v)-1)} + \sum_{v \in com(y_2,z_2)} \frac{2}{d(v)(d(v)-1)}\right].$$

Similarly, when we edges (y_1, y_2) and (z_1, z_2) are added, the clustering coefficient increases by

$$\frac{1}{|G|}\left[\sum_{v \in com(y_1,y_2)} \frac{2}{d(v)(d(v)-1)} + \sum_{v \in com(z_1,z_2)} \frac{2}{d(v)(d(v)-1)}\right].$$

Thus, depending on the relative sizes and degree of the vertices in $com(y_1, y_2) \cup com(z_1, z_2)$ versus $com(y_1, y_2) \cup com(z_1, z_2)$, each step in the process can be relatively small.

Generating random graphs based on our $sparse_{50}$ graph took approximately 9 hours and achieved the desired clustering coefficient of 0.311239 using the BKM approach. Runs with parameters based on the $moderate_{50}$ graph ran for over three days, before we stopped them, and were only able to achieve a clustering coefficient of 0.253527, still quite a ways away from the 0.413189 clustering coefficient of the $moderate_{50}$ graph. These values are given in Table 3.11. Notice that the graphs based on the $sparse_{50}$ graph have equivalent clustering coefficient but only $\frac{1}{4}$ of the triangles of the original graph, suggesting that the triangles formed by the algorithm are incident on low-degree vertices.

Table 3.11: Clustering statistics for graphs generated based on the BKM model.

	Input cc	Output cc	HCW Triangles	Output Triangles
$sparse_{50}$	0.311239	0.311239	313,903	75,269
$moderate_{50}$	0.413189	0.253527$*$	1,948,462	435,629

Shows the input clustering coefficient, the clustering coefficient of the output graph, the number of triangles in the corresponding HCW contact network, and the number of triangles in the output graph.

* indicates that we stopped the generation process prematurely after a number of days.

In Figures 3.16 and 3.17 we see that graphs generated based on the BKM model have clustering coefficient distribution and degree-clustering correlation that is very different from the HCW contact networks. As with the DEG algorithm proposed by Parikh, the mean clustering coefficient of the graphs generated by the BKM model appears to be achieved by having a large number of low-degree vertices with high clustering coefficient. This may again be the result of having a heavy-tailed degree distribution. Since a majority of the vertices in our HCW contact network are of low degree, the edge-swapping algorithm is more likely to choose vertices that are of low degree, and thus, we see that most of the achieved clustering coefficient is due to these low-degree vertices. Further, it may be the case that because the BKM process relies on graphs generated by the CONF model, which has low assortativity, the BKM model cannot efficiently increase clustering coefficient. Recall that for high degree vertex i to achieve high clustering coefficient, there must be a large number of edges between the neighbors of i. Thus, the neighbors of i must also be of higher degree relative to i. But also recall that the CONF model generates unassortative graphs. Thus high degree vertex i may not necessarily be connected to sufficient high degree vertices to provide a higher clustering coefficient.

Finally, comparing disease diffusion curves on graphs generated by the BKM method to the $sparse_{50}$ and corresponding CONF and CA generated graphs, shown in Figures 3.18 and 3.19, we see that the added clustering appears to have no effect on the disease spread. More specifically, disease on the BKM graphs is nearly identical to graphs generated by the CONF model. With low transmissibility, in Figure 3.18, we see similar effect as with the increased clustering coefficient in DEG graphs; there is a minor increase in the disease curve peak as compared with CONF graphs. However, with high transmissibility, in Figure 3.19, there is no noticeable change in the disease curve compared to the CONF graph. This is all despite the BKM graphs having a higher clustering coefficient compared to the DEG graphs. This may suggest that at low transmissibility, clustering may play an important role in disease diffusion compared to high transmissibility. But the results here are only considering networks which have increased clustering around low degree vertices and since high degree vertices are left relatively unchanged, there is little difference in the behavior of disease diffusion.

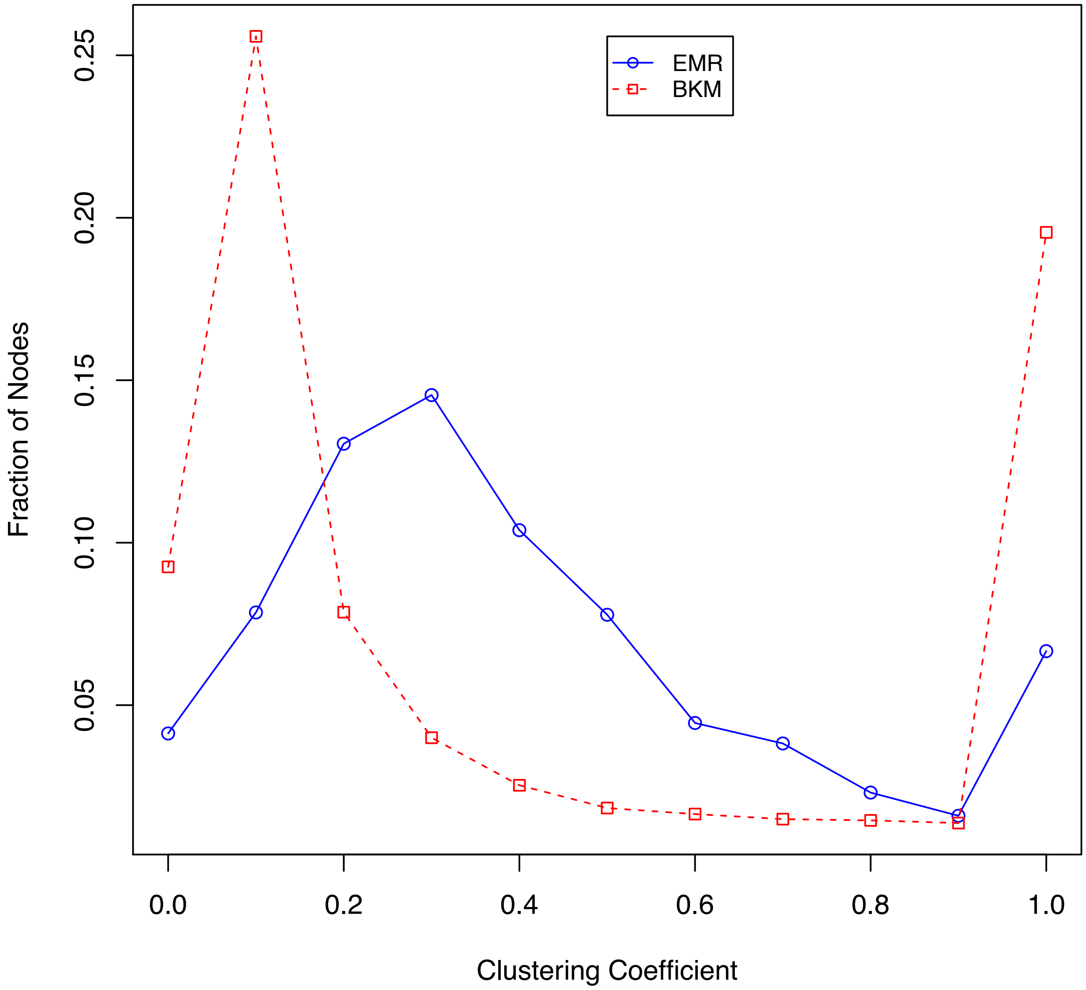


Figure 3.16: Clustering coefficient distribution for $sparse_{50}$ graph and corresponding graphs generated by the BKM model. Vertices with degree less than 2 are ignored.

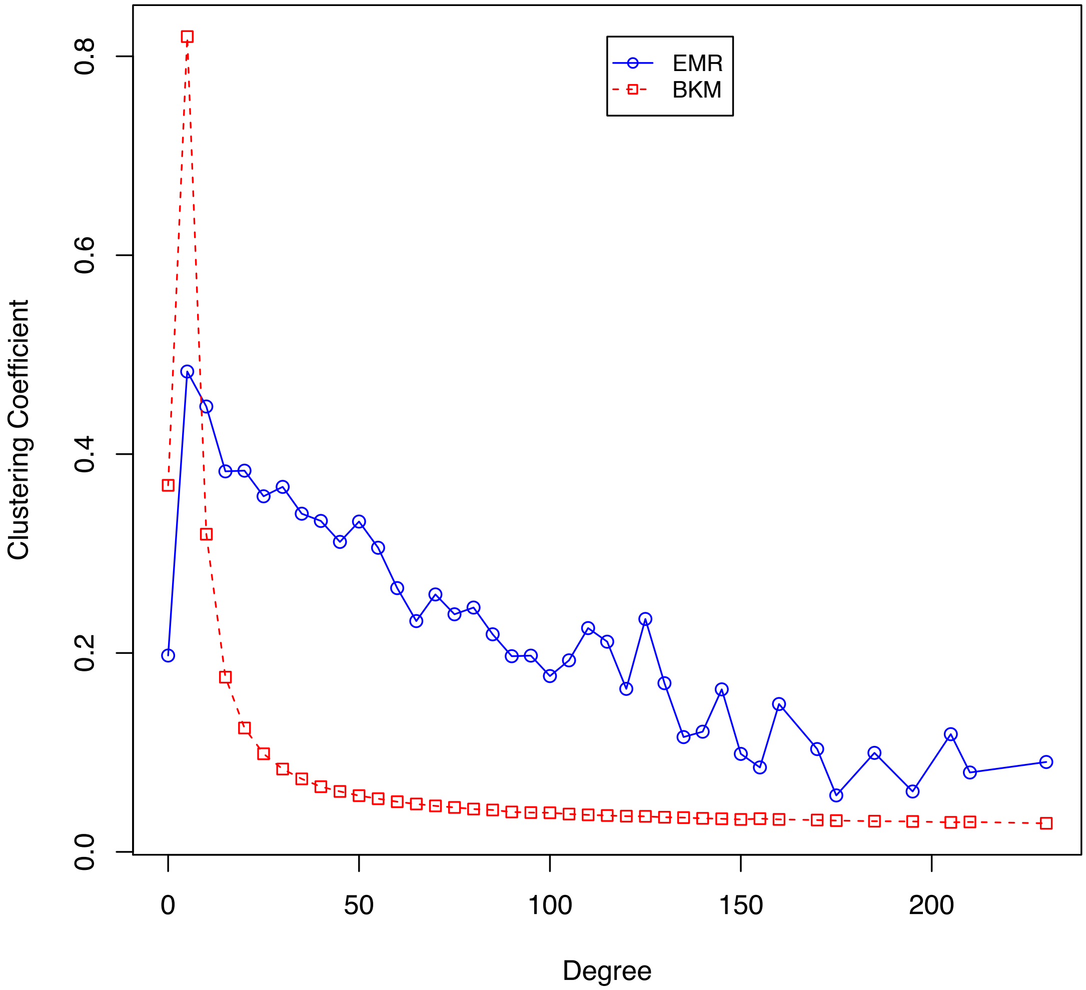


Figure 3.17: Vertex degree vs clustering coefficient correlation plot for $sparse_{50}$ graph and corresponding graphs generated by the BKM model. Vertices are grouped into bins based on degree by groups of 5.

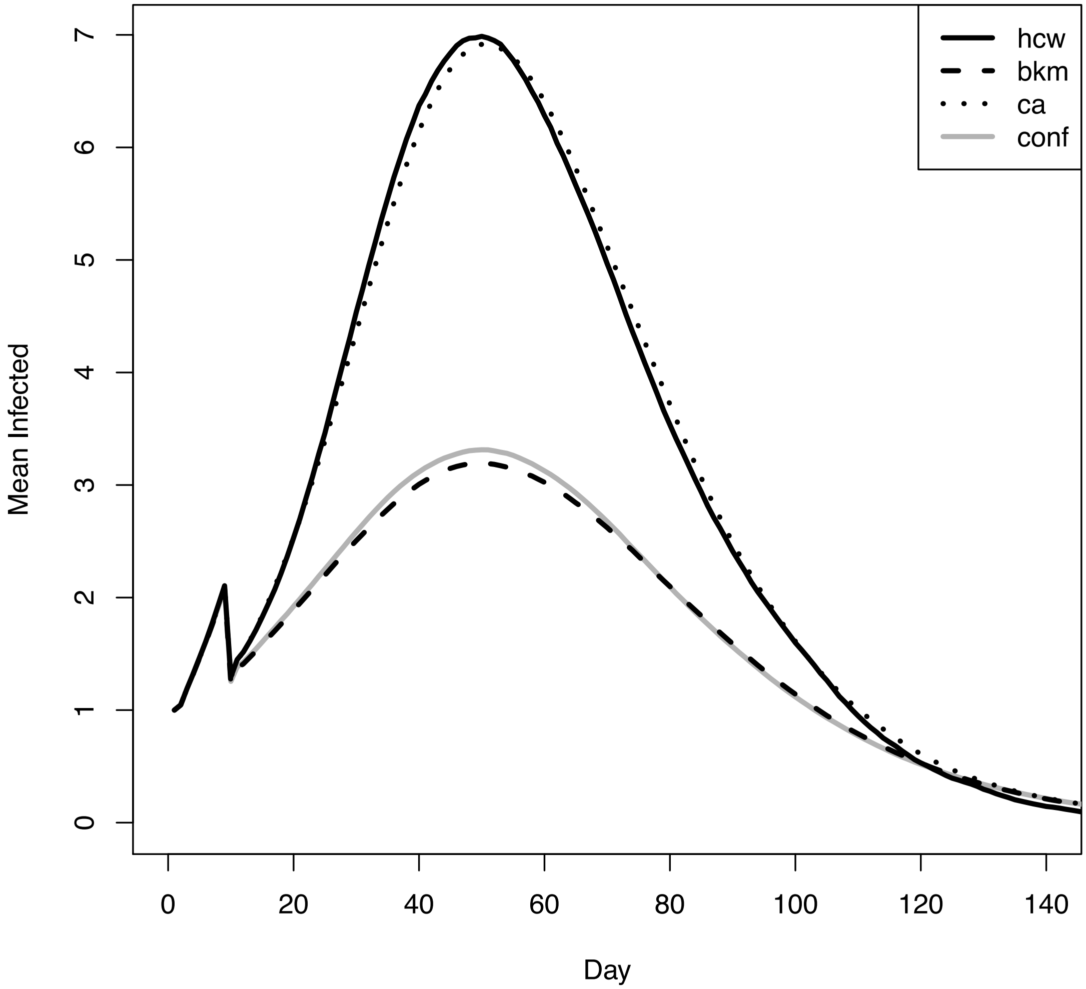


Figure 3.18: Comparison of $sparse_{50}$ graph with graphs generated using the BKM, CONF, and CA models with $\rho = 0.0122$.

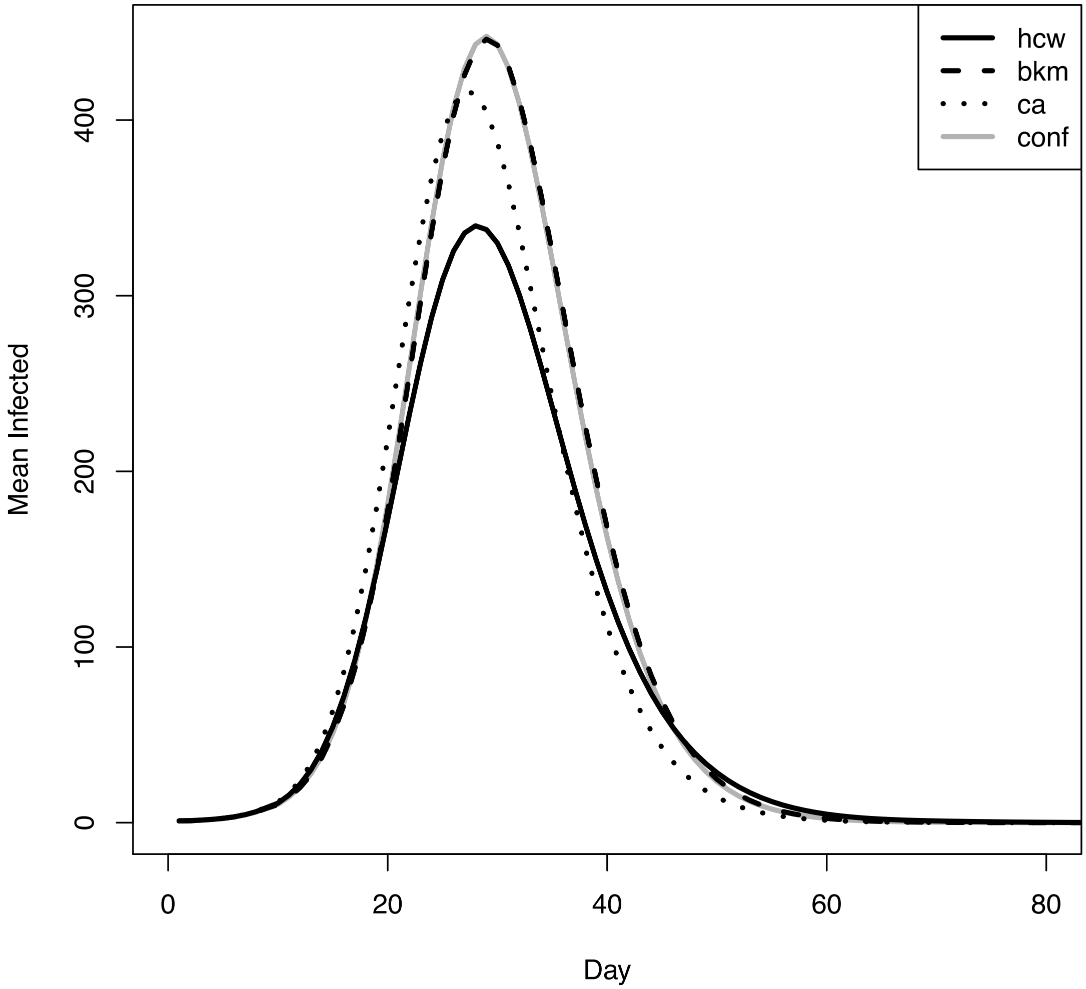


Figure 3.19: Comparison of $sparse_{50}$ graph with graphs generated using the BKM, CONF, and CA models with $\rho = 0.048$.

3.6 Spatial-Clustering Model

We introduce a simple model for generating random graphs with clustering that we call the Spatial-Clustering (SC) model. In this model each vertex is assigned a point on a euclidean plane and a graph is constructed in a manner similar to the configuration model but with a bias towards connecting stubs that are spatially close.

Random graphs described by the SC model are defined by a degree sequence $(d_1, d_2, \ldots, d_n)$ and parameter γ. The SC model starts with n vertices with each vertex v having d_v "open" stubs. Each vertex v is assigned a location $l_v = (x, y)$ in Euclidean space such that $0 \leq x \leq 1$ and $0 \leq y \leq 1$. For two vertices u, v let $d(u, v)$ denote the Euclidean distance between the locations assigned to each vertex. Let $S(v)$ denote the set of open stubs for vertex v. The SC model constructs random graphs by placing edges between pairs of open stubs for vertices u, v in the following manner. For a vertex u, picked uniformly at random from all open stubs, a stub adjacent to vertex v is picked with probability proportional to

$$\frac{S(v)}{\sum_w S(w)} \cdot \frac{1}{d(u,v)^\gamma}.$$

Table 3.12 shows statistics for graphs generated by the SC model and the HCW contact networks as well as the CONF and CA models. It is interesting to note that all three random graph models match the exact degree distribution and graphs generated from the SC and CA models only differ from the CONF model by clustering coefficient and assortativity respectively.

It turns out that the graphs generated by the SC model have a very similar clustering coefficient distribution and degree-clustering correlation as shown in Fig-

Table 3.12: Graph statistics for the *moderate*$_{50}$ graph and graphs generated Spatial-Clustering (SC) with $\gamma = 3$, CONF, and Configuration with Assortativity models.

	HCW	SC	CONF	CA
n	7144	7144.0	7144.0	7144.0
m	191270	191270.0	191270.0	191270.0
$\langle k \rangle$	53.547	53.547	53.547	53.547
σ	64.704	64.704	64.704	64.704
k_{max}	660	660.0	660.0	660.0
r	0.139	-0.008 (±0.002)	-0.016 (±0.001)	0.139
cc	0.413	0.433 ±0.002	0.041 (±0.001)	0.03
t	0.234	0.214 ±0.001	0.043	0.062
c	323	300.0	300.2 (±0.4)	315.167 (±0.373)
$singletons$	299.0	299.0	299.0	299.0
n_{giant}	6770	6845.0	6844.55 (±0.921)	6814.5 (±1.118)
m_{giant}	191196	191270.0	191269.75 (±0.536)	191254.667 (±0.745)
$\langle \ell \rangle$	3.134	2.916 (±0.003)	2.666	2.829
$diam$	11	6.167 (±0.373)	5.85 (±0.357)	8.3 (±0.458)

For random graph models values are averaged over 20 generated graphs and ± values indicate significant (> .0001) standard deviation.

ures 3.20 and 3.21. One theory about why the SC model is so successful is related to the uniform distribution of clustering coefficient of the HCW contact networks. Since the SC model distributes vertices uniformly on a 2D plane and there is a bias or picking pairs of stubs that are close, vertices within a given area are likely to be densely connected. Consider three vertices u, v, w such that there exists an edges (u, v) and (v, w). Since these edges exists, it suggests that u, v and w are all spatially close, and thus u, w must also be spatially close and edge (u, w) is also likely to exist. By distributing vertices uniformly on the plane, all vertices are likely to have roughly the same clustering coefficient.

Figures 3.22 and 3.23 show disease diffusion curves for the $moderate_{50}$ graph with low and high transmissibility levels respectively. While the generated graphs for the SC have clustering coefficient and distribution that is very close to the HCW contact networks, it turns out that the disease curves are more like the graphs generated by the CONF model. The squared distances between the curves, given in Tables 3.14 and 3.16, reflect this. However, one interesting aspect of the SC graphs is that the disease peak is lower and lags slightly behind the CONF graphs, despite infecting roughly the same number of people. It seems that the introduction of uniform clustering slows the spread of disease without reducing it. This is likely due to fewer "long-distance" edges, since vertices are much less likely to be connected to other vertices that are spatially far away.

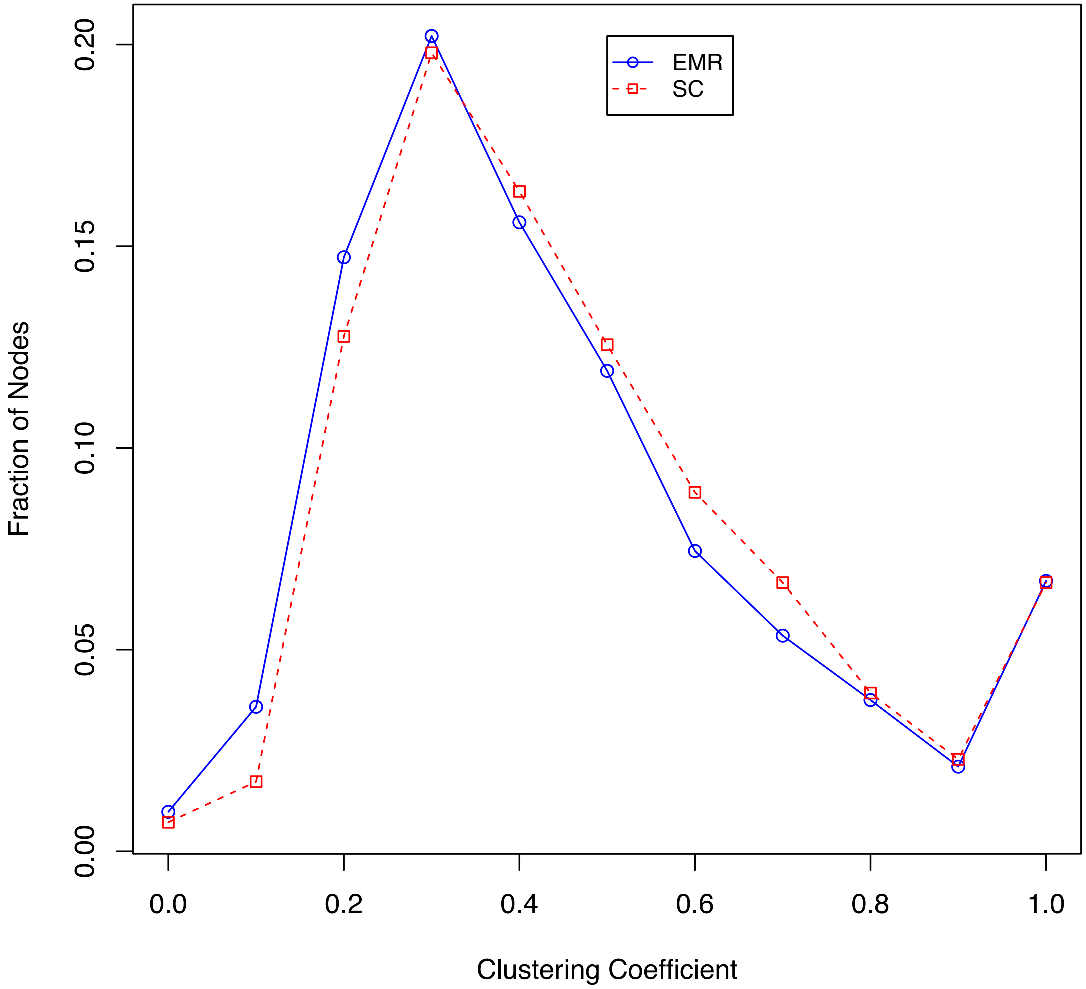


Figure 3.20: Clustering coefficient distribution for $moderate_{50}$ graph and corresponding graphs generated by the SC model. Vertices with degree less than 2 are ignored.

Table 3.13: Statistics for the plot in Figure 3.22 showing mean and median number of individuals infected and the mean number of new infections each day.

	HCW	SC	CA	CONF
Mean Inf.	317.291	322.853	328.448	332.196
Median Inf.	1	1	1	1
Mean $+\Delta$	0.229	0.163	0.248	0.176
Peak Day	39	47	40	45

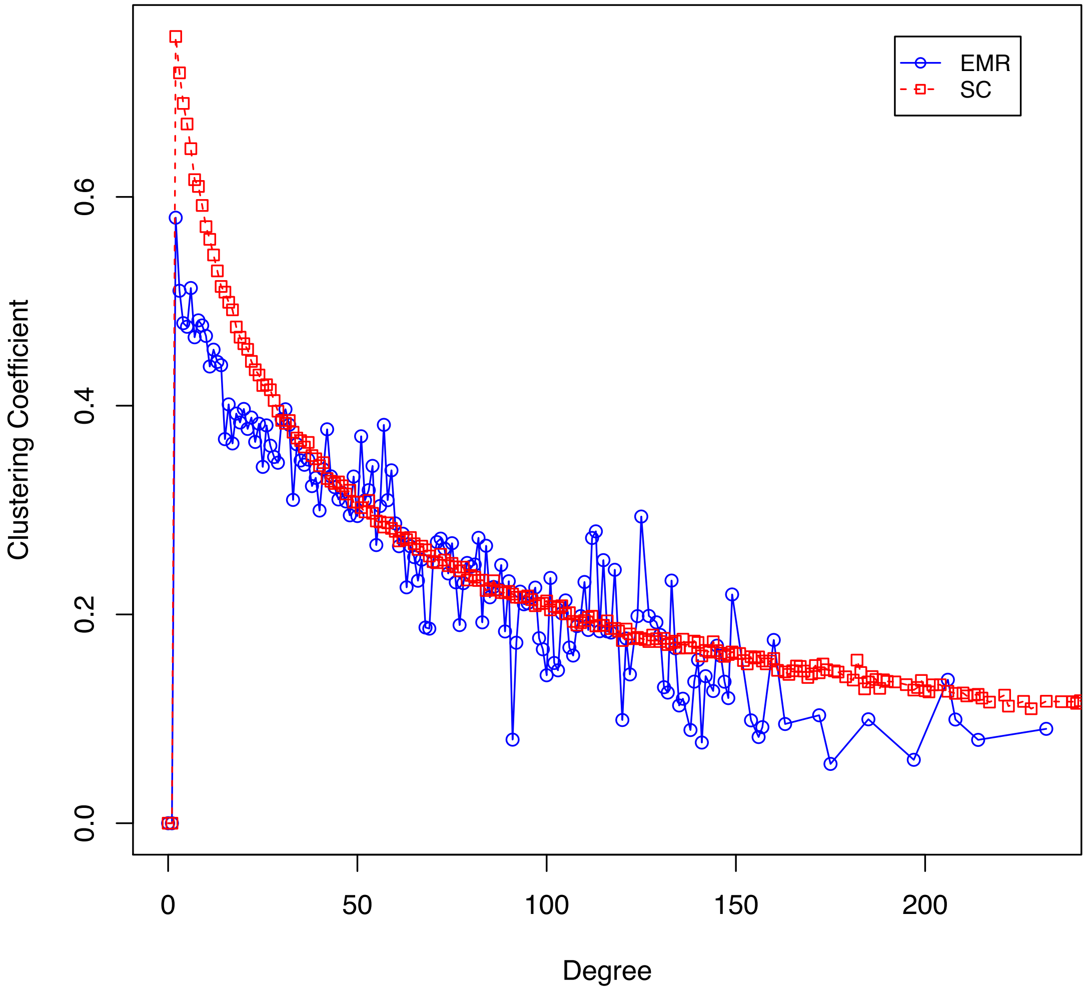


Figure 3.21: Vertex degree vs clustering coefficient correlation plot for $moderate_{50}$ graph and corresponding graphs generated by the SC model.

Table 3.14: Numerical comparison of the curves in Figure 3.22 measured by the sum of the squared distances each day.

	HCW	SC	CA	CONF
HCW	0.00000	164.90980	33.11518	143.08838

Smaller values indicate more similarity

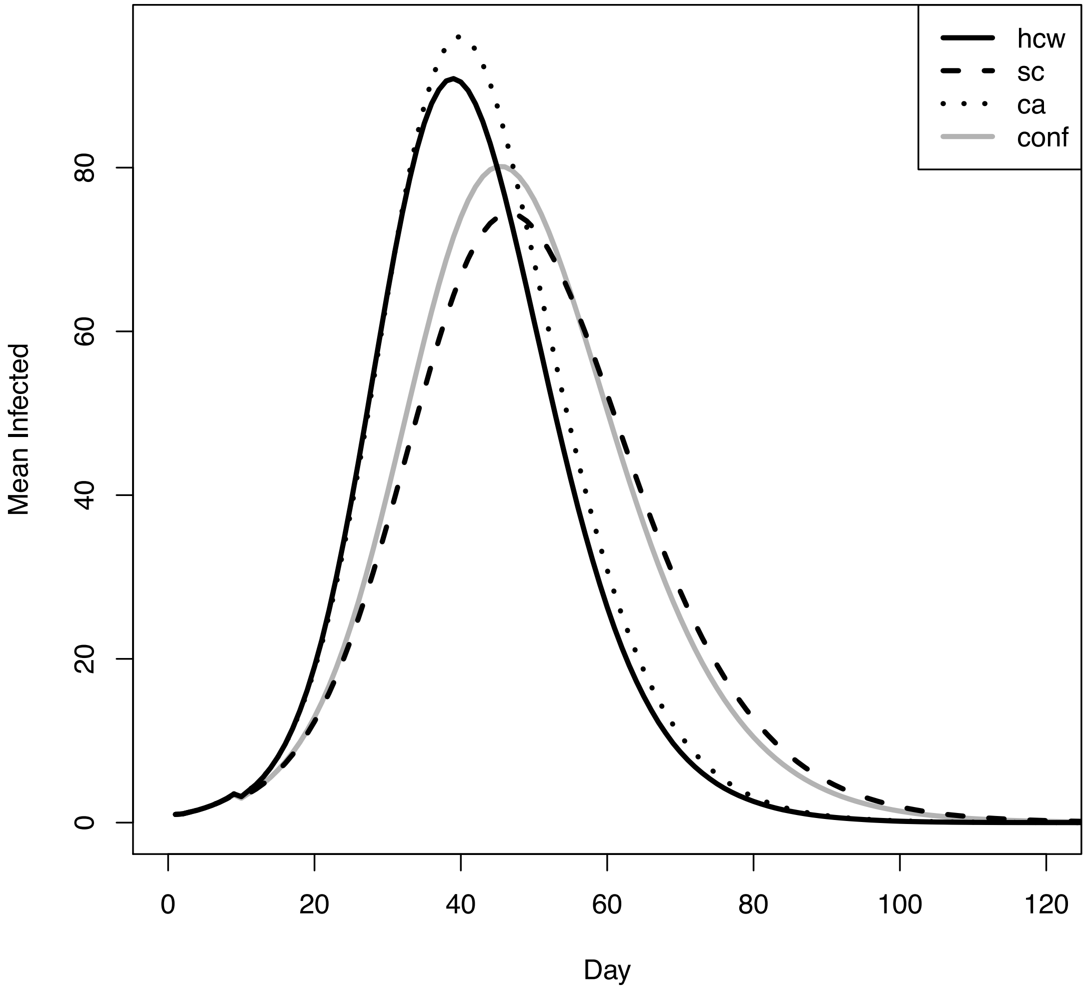


Figure 3.22: Number of people infected on each day of an SIR simulation with low transmissibility $\rho = .0122$ for the $moderate_{50}$ graph and the Spatial-Clustering (SC), Configuration (CONF), and Configuration with Assortativity (CA) models. x-axis represents the timestep for the simulation (days). y-axis gives the number of people in the "infected" state during that timestep.

Table 3.15: Statistics for the plot in Figure 3.23 showing mean and median number of individuals infected and the mean number of new infections each day.

	HCW	SC	CA	CONF
Mean Inf.	1697.460	2013.200	1795.006	2017.214
Median Inf.	3	3739	3460	3739
Mean $+\Delta$	5.997	6.941	8.145	8.002
Peak Day	22	24	22	23

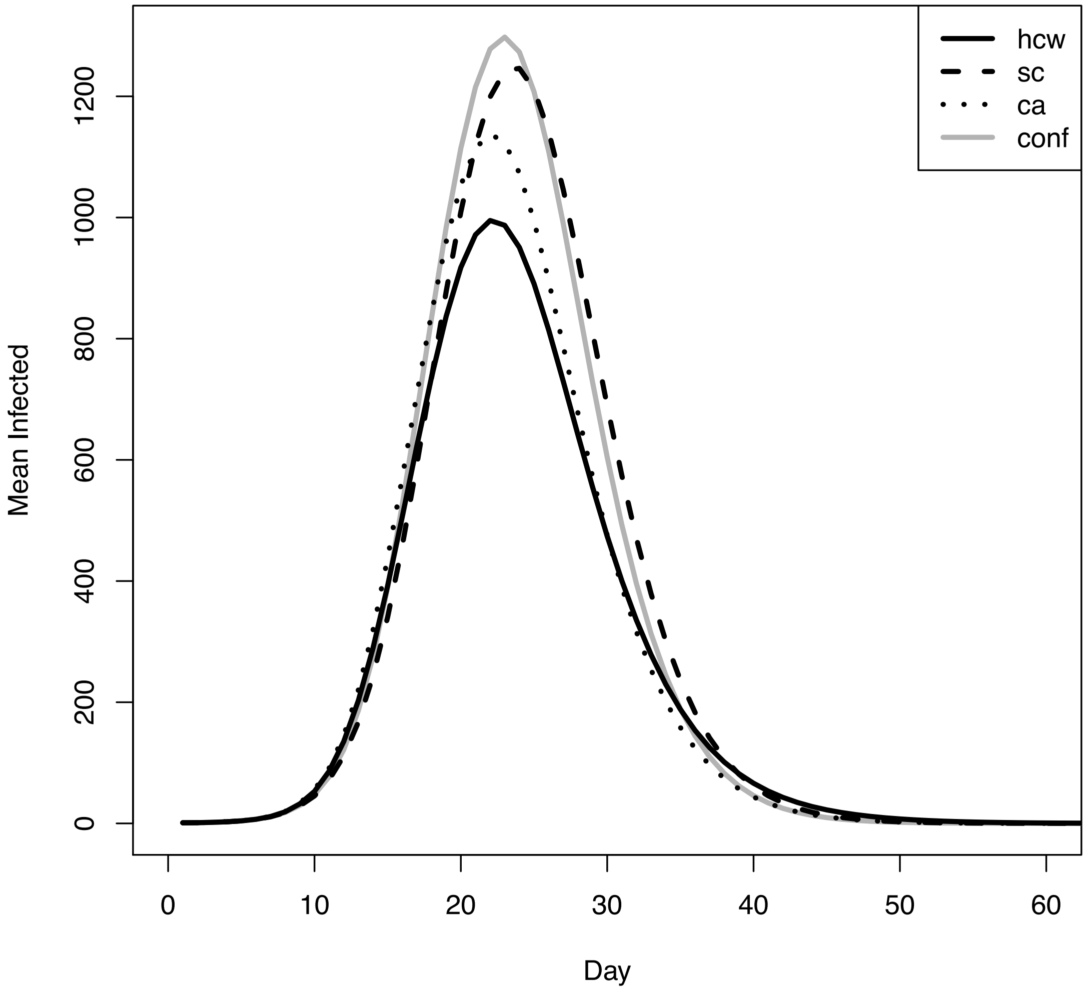


Figure 3.23: Number of people infected on each day of an SIR simulation with low transmissibility $\rho = .122$ for the $moderate_{50}$ graph and the Spatial-Clustering (SC), Configuration (CONF), and Configuration with Assortativity (CA) models. x-axis represents the timestep for the simulation (days). y-axis gives the number of people in the "infected" state during that timestep.

Table 3.16: Numerical comparison of the curves in Figure 3.23 measured by the sum of the squared distances each day.

	HCW	SC	CA	CONF
HCW	0.0000	900.1287	405.5003	883.2523

Smaller values indicate more similarity

3.7 Discussion

These results suggest that not only is positive assortativity an important characteristic of the HCW contact networks we generate, but also that the clustering coefficient in these graphs may not be as important. And since the HCW contact networks that we generate have properties similar to other real-world contact networks, these results may have implications for diffusion processes on networks that arise in other contexts.

We hypothesize that positive assortativity plays a key role in the outcome of disease spread because that there is less mixing of connections in graphs with high assortativity. The lack of mixing in the HCW contact networks and CA graphs is suggested by the fact that these assortative graphs have a larger diameter, as compared with the CONF and SC graphs. At lower transmissibility this lack of mixing helps disease survive and inevitably infect a larger fraction of the population. For example, consider that in an assortativity graph, when a high degree vertex becomes infected, it is more likely to pass that disease on to another high degree vertex than a low degree vertex. When transmissibility is low, this correlation between neighbors makes the disease more likely to continue on between vertices of high degree. At high transmissibility this has the opposite effect, as these few high degree vertices are densely connected in order to achieve the given assortativity and thus disease spread tends to get suck in this densely connected "pocket" of high degree vertices. On the other hand, graph models that incorporate clustering coefficient don't appear to limit the level of mixing as much. And at high transmissibility this mixing allows disease to quick spread to a large part of the population.

CHAPTER 4
VACCINATION POLICIES

4.1 Vaccination Problem

We model disease diffusion as a dynamical process on a network in which, given an initial set of infected individuals (possibly chosen at random), disease spreads according to a diffusion model M. In general, diffusion models describe a stochastic or deterministic process of how some “thing” (information, disease, etc.) spreads within a population. For our work we focus on stochastic diffusion models, which we will refer to as *disease diffusion* models, that attempt to approximate the spread of disease on social contact networks. These disease diffusion models follow the trends of SIR-like models [50] where each individual is in one of the following states: “susceptible” to infection, “infected” with the given disease, or “recovered” from infection and immune to further infection. The disease diffusion model describes behavior by which infection spreads from infected individuals to susceptible individuals and how infected individuals eventually recover from the disease. The model used in the previous chapter to compare random graph models is an example of one of the types of disease diffusion models in which we are interested.

For a diffusion model M and graph $G = (V, E)$ we let $I_M(G)$ denote a random variable that gives the the number of people that contract the disease, which spreads according to M, starting with a single infected individual chosen uniformly at random from V. For a random variable X we denote with $E[X]$ the expected value of X. Also, we denote $G \setminus V'$ as the subgraph resulting from removing vertex set $V' \subseteq V$ from graph G. Thus, $E[I_M(G \setminus V')]$ denotes the expected number of individuals that get infected starting with a single infected individual in $V \setminus V'$ and spreading according to M. This models the process of vaccinating a set of individuals, in this case V', and starting disease at random from any unvaccinated individual.

Informally speaking, there seem to be two problems relating to vaccination from an application point of view. The first is that there is a limited supply of vaccines available and we want to efficiently vaccinate as to minimize the spread of disease. The second is that we would like to limit the spread of disease but would like to minimize the cost (number of vaccinations) of doing so. Both of these problems can be abstracted as optimization problems; given some budget we want to optimize some objective function. In general, as one can imagine, the outcome of some epidemic has a lot of confounding factors which makes the *vaccination problem* very difficult to solve.

4.2 Budgeted Vaccination Problem

We first consider the *budgeted vaccination problem* which aims to minimize the number of infected individuals by strategically vaccinating some subset of individuals.

Budgeted Vaccination problem (BV): Given a contact network $G = (V, E)$ with n vertices, positive integer budget B, and diffusion model M, find

$$\arg\min_{V' \subseteq V} E[I_M(G \setminus V')] \qquad \text{s. t.} \quad |V'| \leq B.$$

The vaccination problem can be re-framed as a maximization problem by defining the function $f : 2^V \to Z^+$ as

$$f(V') = n - E[I_M(G \setminus V')]. \tag{4.1}$$

The maximization version of the vaccination problem is then,

$$\arg\max_{V' \subseteq V} f(V') \qquad \text{s. t.} \quad |V'| \leq B.$$

In more general terms, given the diffusion model M, the maximization version of the vaccination problem is to pick a set of B individuals that maximize, on average, the number of people that remain *uninfected* by the disease after it infects a single individual uniformly at random. Notice that an optimal solution for the maximization version of the vaccination problem is an optimal solution the minimization version and vice versa.

Budgeted maximization problems like BV are NP-hard in general [78]. For budgeted maximization of *submodular* functions there are well-known approximation algorithms which provide a $\left(1 - \frac{1}{e}\right)$ factor approximation [78, 20, 32, 55]. An algorithm A is an α-approximation, $0 < \alpha < 1$, for a maximization problem if the cost of the solution returned by A is at least $\alpha \cdot OPT$ where OPT is the cost of the optimal solution. A function $f : 2^S \to R$ is submodular iff

$$f(A \cup \{a\}) - f(A) \geq f(B \cup \{a\}) - f(B)$$

for all $A \subseteq B \subseteq S$ and $a \in S \setminus B$.

Unfortunately the objective function for the BV is not submodular. For example, consider a disease diffusion model M where disease spreads across an edge in

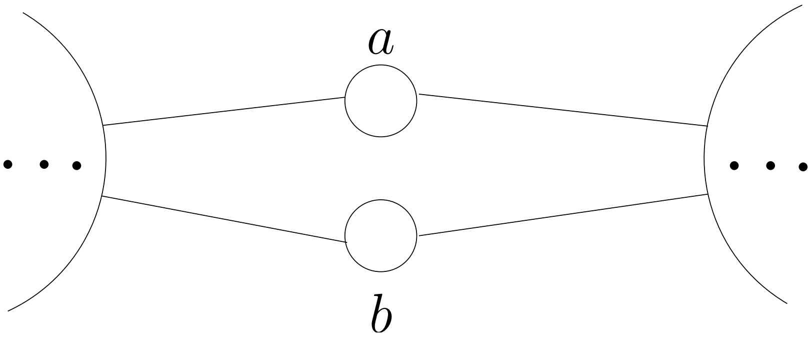


Figure 4.1: Simple graph of two vertices between two components of equal size. Vaccinating b alone gives really no improvement but vaccinating b along with a gives a major improvement by separating halves of the graph. For the *budgeted maximization problem* this demonstrates the objective function f violates the submodular property of "diminishing returns."

the contact network with probability 1. Let G be the graph shown in Figure 4.1 with two vertices, a and b, which connect two connected components of size $(n-2)/2$. Initially G is a single connected component of size n. With f defined as in (4.1) we have,

$$f(\emptyset) = 0$$

$$f(\{a\}) = 1$$

$$f(\{b\}) = 1$$

$$f(\{a,b\}) = \frac{n}{2} + 1$$

Since BV is difficult in general, primarily due to the dynamic nature of the diffusion process, a number of "proxy" problems which abstract away some of the complexity have been considered by researchers. The "proxy" problems themselves turn out to be NP-hard and can only be solved via approximation algorithms or heuristics. More importantly, these "proxy" problems turn out to be reasonable proxies only under limited circumstances, as we will show later.

4.2.1 Sum-of-Squares Partition Problem

Related literature on the problems of vaccination on contact networks make the assumption that disease spreads in a worse-case fashion to all vertices in the connected component of the initial infected individual [7, 26]. With this in mind, Aspnes et al. [7] consider the *sum-of-squares partition* problem, derived from problems in network security, which aims to minimize the sum of the squares of component sizes in a network. Consider the worst case disease diffusion model on a graph G with connection components $H_1, H_2, \ldots, H_l$ starting with an initial infected individual chosen uniformly at random. The expected number of people infected is

$$\frac{(H_1)^2}{n} + \frac{(H_2)^2}{n} + \cdots + \frac{(H_l)^2}{n}.$$

Based on this, given a budget of vaccinations, the intelligent choice is to try and pick vertices in order to minimize the numerator; the sum-of-squares of the component sizes. This is exactly the goal of the *sum-of-squares partition* problem.

Sums-of-Squares Partition problem: Given a graph $G = (V, E)$ and budget B, find $V' \subset V$ of at most B vertices where removing V' from G leaves disconnected components $H_1, H_2, \ldots, H_l$ such that $\sum_i |H_i|^2$ is minimized.

Suppose that an optimal solution for the sum-of-squares partition problem with budget B finds a partition of G into components of size $h_1, h_2, \ldots, h_j$. Aspnes et al. [7] provide a polynomial time approximation algorithm that can find a set of $O(\log^{1.5} n)B$ vertices that partition the graph into components $h_1^{'}, h_2^{'}, \ldots, h_k^{'}$ such that $\sum_k h'_k \leq O(1)\sum_j h_j$. This leads to an $O(\log^{1.5} n)$-approximation for the *sum-of-squares* problem [7].

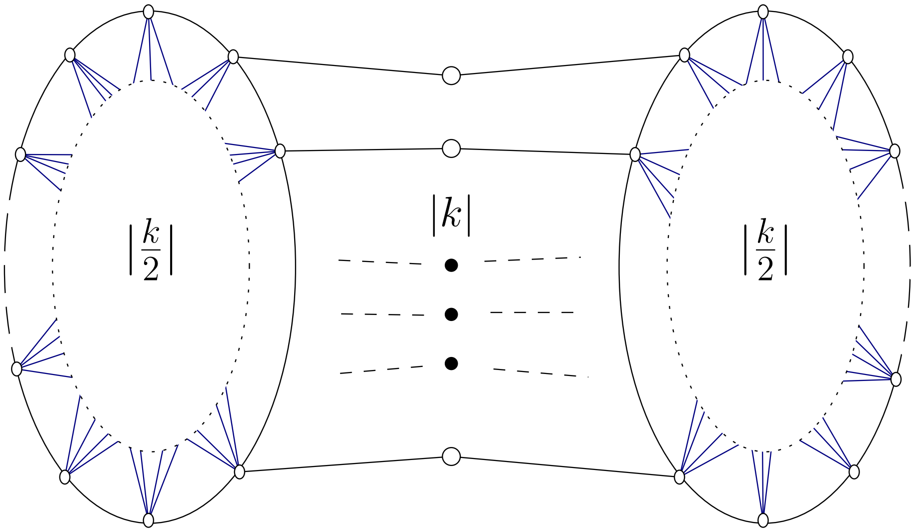


Figure 4.2: Instance of BV for which sum-of-squares partition problem is a poor proxy.

It is clear, or soon will be, that the sum-of-squares partition problem doesn't always provide "good" solutions to the vaccination problem. The goal of the sum-of-squares problem is to partition the graph into components but it makes no guarantees about the density of these connected components and does not take the disease diffusion model into consideration. Consider the example graph of n vertices in Figure 4.2. There are two components separated by k vertices where $n \gg k$. Each component is composed of a ring of $\frac{n-2k}{2}$ vertices and a clique of $\frac{k}{2}$ vertices where each vertex in the ring is connected to all vertices in the clique. For a budget of k, an optimal solution to the sum-of-squares partition problem is the k vertices that separate the two components.

Let OPT_s be the optimal solution to the sum-of-squares partition problem and let S_b be the set of k vertices that make up the two $\frac{k}{2}$ cliques in each connected component. Now consider a disease diffusion model M that spreads across edges at each time unit with probability $\frac{1}{\sqrt{k}}$ and let R_0 denote the number of infections after the first time unit (i.e., the number of secondary infections spread from the initially infected individual chosen uniformly at random). The following lemmas hold for graph G in Figure 4.2.

Lemma 4.1. *For graph $G \setminus OPT_s$ and disease diffusion model M, $R_0 \geq \frac{\sqrt{k}}{2}$.*

Lemma 4.2. *For graph $G \setminus S_b$ and disease diffusion model M, $R_0 \leq \frac{3}{\sqrt{k}}$.*

Lemma 4.1 follows by the following argument. If we remove OPT_s from the graph G we effectively split the graph into two connected components of size $\frac{n-k}{2}$. The minimum degree in the resulting graph is bounded below by $\frac{k}{2}+2$. By M, disease will be transmitted across each edge with probability $\frac{1}{\sqrt{k}}$. Thus $R_0 \geq \frac{1}{\sqrt{k}}\frac{k}{2} = \frac{\sqrt{k}}{2}$.

By similar argument, if we remove S_b from the graph G we remove the cliques in each of the two connected components. Vertices in the resulting graph have degree bounded above by 3. Thus $R_0 \leq \frac{3}{\sqrt{k}}$ and Lemma 4.2 holds.

By Lemmas 4.1 and 4.2, after one time unit, the cost of the sum-of-squares partition solution is a $\frac{3}{2}$ approximation for the minimization version of BV.

4.2.2 Budgeted Vertex Cover

A second "proxy" for BV is the *budgeted vertex cover* problem that aims to reduce the average degree of the graph.

For a graph $G = (V, E)$ define the function $e^-_G : 2^V \rightarrow Z^+$ as the number of edges in the graph $G \setminus V'$ for $V' \subseteq V$.

Min Budgeted vertex cover (Min-BVC): Given graph G and budget B find a subset $V' \subseteq V$ of size no greater than B that minimizes $e^-_G(V')$.

In general it is difficult to measure the quality of the solution returned by approximation algorithms for this problem due to the fact there are instances of the problem with optimal value $OPT = 0$. Any approximation algorithm that returns a solution S, where $e^-_G(S) > 0$, will have an approximation ratio of infinity if $OPT = 0$.

To fix this problem we can reframe Min-BVC as a maximization problem. Define the function $e^+_G : 2^V \rightarrow Z^+$ as the $e^+_G(V') = |E| - e^-_G(V')$. The maximization version of Min-BVC is then to maximize the function e^+_G over all subsets $V' \subseteq V$, $|V'| \leq B$. We call this problem Max-BVC and note that $e^+_G(V')$ is the total number of edges incident on vertices in V'.

It turns out that e^+_G is a *submodular* set function. Thus, instances of the budgeted vertex cover problem can be reduced to instances of the *budgeted maximum coverage* problem [55]. For the budgeted maximum coverage problem the simple greedy algorithm, which we will refer to as **GreedyBVC**, gives a $(1 - 1/e)$ approximation. In terms of BVC, this means that for any graph, **GreedyBVC** will give a solution that covers over 63% of the edges covered by the optimal solution. Further, Khuller et al. [55] show that this bound is the best that can be achieved in polynomial time.

GreedyBVC starts with $V' = \emptyset$ and iteratively adds to V' the vertex, $v \in V \setminus V'$, which maximizes $e^+_G(V' \cup \{v\}) - e^+_G(V')$ until $|V'| = B$. This greedy process is effectively picking the vertex v with *highest degree*, adding that to the solution set, removing v from the graph, and repeating the process.

Consider an instance of Max-BVC on the graph G in Figure 4.2 with budget $B = k$. The optimal solution to the Max-BVC problem on G is set of k vertices in the two $k/2$ cliques, denoted as OPT_{bvc}. For an instance of BV on graph G with budget k and disease diffusion model M where disease spreads across edges with probability $\frac{1}{\sqrt{k}}$, the optimal solution to BV is exactly OPT_{bvc}. It also happens to be the solution returned by the **GreedyBVC** algorithm.

However, it is not always the case that Max-BVC will return an optimal solution to the vaccination problem. Consider an instance of BV on graph G, from Figure 4.2, with budget k and an aggressive disease diffusion model M' that spreads to all vertices in the connected component where infection starts. The optimal solution to BV is the k vertices between the two components with an expected number of infected of $\frac{n-k}{2}$. However, OPT_{bvc} has expected number of infected of $n - k$, and thus, for this instance, Max-BVC provides a 2 approximation for BV.

For instances of Max-BVC on our HCW contact networks we experimentally test the quality of **GreedyBVC** by comparing it with solutions of the corresponding linear program (LP). The LP is relaxed from the IP, defined below, and gives us an upper bound on the optimal integer solution. An instance of Max-BVC $\{G = (V, E), B\}$ can be realized as an Integer Program (IP) by assigning a variable x_u to each $u \in V$ such that $x_u = 1$ iff $u \in V'$. We assign y_e to each edge $e \in E$ where $y_e = 1$ iff e is removed from G by dropping vertices in the solution V'. The IP is,

$$\max \sum_e y_e$$

$$\begin{aligned} \text{subject to} \quad & x_u + x_v \geq y_e & & \forall (u, v) = e \in E \\ & \sum_u x_u \leq B & & \\ & x_u \in \{0, 1\} & & \forall u \in V \\ & y_e \in \{0, 1\} & & \forall e \in E. \end{aligned}$$

We can relax this to a linear program (LP) simply by relaxing the final two constraints so that $0 \leq x_u \leq 1$ and $0 \leq y_e \leq 1$.

Experimental runs of budgeted vertex cover problem perform much better on our HCW contact networks than is guaranteed by the approximation. Figure 4.3 shows a plot comparing the greedy algorithm for budgeted maximum coverage with an upper bound found by solving the LP relaxation. For small budgets of vertices (less than .1 fraction of the vertices in Figure 4.3) and large budgets of vertices (greater than a .65 fraction of the vertices in Figure 4.3), the greedy finds the optimal solution. When the greedy fails to achieve the lower bound, we measure the quality of the greedy solution over the quality of the solution returned by the LP. This gives us an experimental approximation ratio for the greedy solution. For the plot in Figure 4.3, the worst approximation ratio, for budgets where greedy doesn't achieve the lower

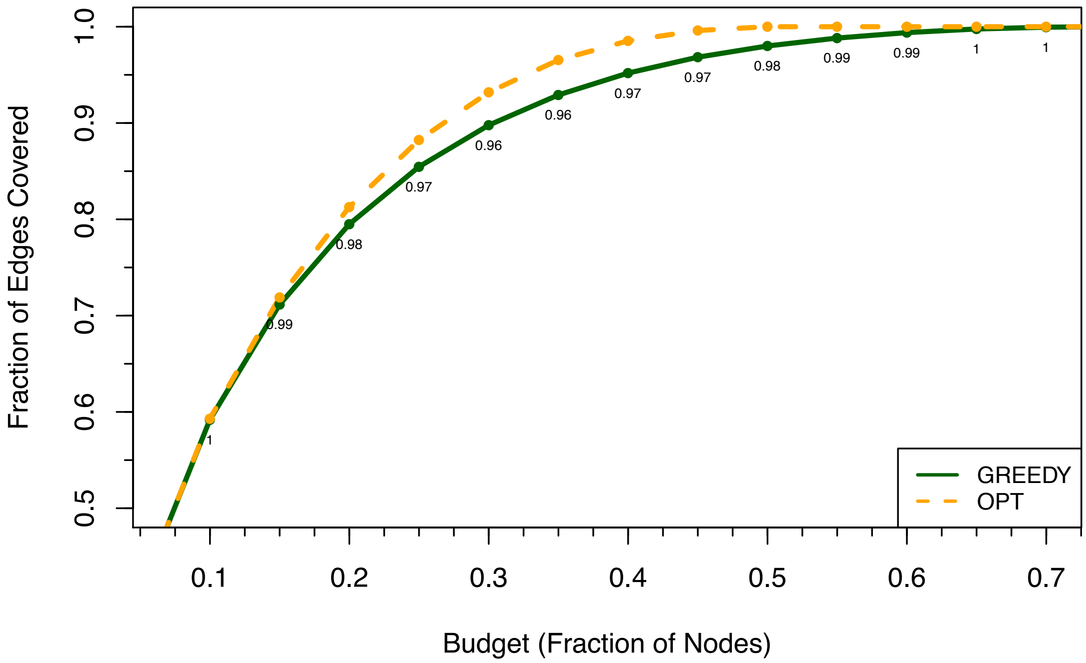


Figure 4.3: Comparison of experimental runs on the *moderate*$_1$ HCW contact networks for the greedy budgeted vertex cover algorithm and an upper bound calculated by solving the LP relaxation. The x-axis is truncates values that have the same y values.

bound, is $.96(\frac{24}{25})$. These experiments suggest that if solving BVC is appropriate for graphs like our HCW contact networks, then the simple greedy is providing a near-optimal solution to the vaccination problem. This would imply that vaccinating well-connected individuals is the correct policy for vaccination.

4.2.3 Min-Max Degree

The *budgeted vertex cover* problem is effectively minimizing the average degree. Another reasonable "proxy" problem that we consider takes the approach of minimizing the maximum degree based on the intuition that this will minimize the probability that disease will spread away from a given vertex.

For graph $G = (V, E)$ let the function $h_G^- : 2^V \rightarrow Z^+$ be the maximum degree of any vertex in the graph $G \setminus V'$ for $V' \subseteq V$.

Budgeted Min-Max Degree (BMD): Given graph $G = (V, E)$ and budget B, find $V' \subseteq V$ of size no greater than B that minimizes $h_G^-(V')$.

The maximization version of BMD is to maximize the function $h_G^+ : 2^V \rightarrow Z^+$ defined as $h_G^+(V') = \Delta(G) - h_G^-(V')$ where $\Delta(G)$ is the maximum degree of any vertex in G and $V' \subseteq V$.

Like the Budgeted Vaccination problem, the object function h_G^+ for the maximization version of BMD is not submodular. Consider the example graph in Figure 4.4

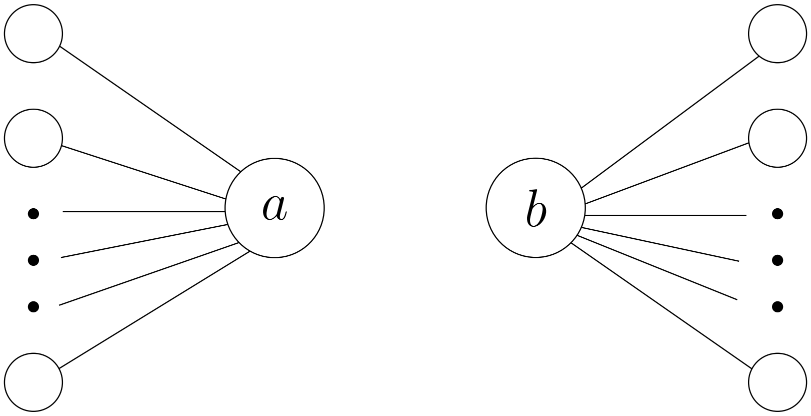


Figure 4.4: Example graph showing that h_G+ is not submodular.

where we have,

$$
\begin{aligned}
h_G^+(\emptyset) = & \quad 0 \\
h_G^+(\{a\}) = h_G^+(\{b\}) = & \quad 0 \\
h_G^+(\{a\} \cup \{b\}) = & \quad \tfrac{n-2}{2}.
\end{aligned}
$$

Worse news about BMD is that the natural greedy algorithm is to continually add to the solution set the vertex which maximizes the decrease in the maximum degree. This is problematic because at each iteration of the greedy algorithm, there are times when removal of a single vertex will not decrease the maximum degree. Thus a vertex has to be removed from the set of all vertices arbitrarily. An alternate greedy algorithm would be to adopt the greedy from BVC and continually pick the vertex of maximum degree. This algorithm is at least twice as bad as the optimal solution as shown by the bad example in Figure 4.5.

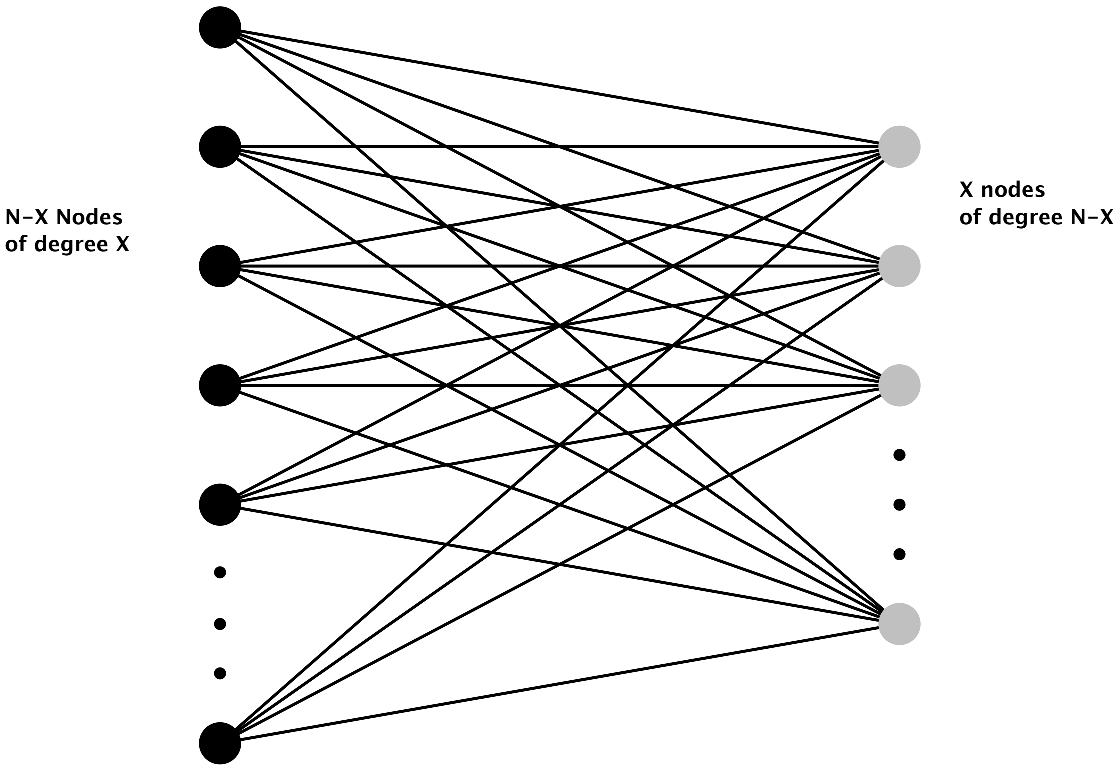


Figure 4.5: Bad Case for BMD. Suppose that $B = X - 1$ and $X = \frac{N}{3}$ so that each gray vertex has degree $\frac{2N}{3}$ and each black vertex has degree $\frac{N}{3}$. The optimal solution would be B black vertices resulting in reducing the maximum degree to $\frac{N}{3} + 1$. A greedy solution based on degree would pick B of the gray and the maximum degree would remain $\frac{2N}{3}$. The greedy algorithm is thus *at least* a 2 approximation.

4.3 Restricted Disease Problem

In addition to BV we consider its "dual" which we call the *Restricted Disease* problem. Roughly speaking, the Restricted Disease problem attempts to minimize the number of people we need to vaccinate in order to keep disease outbreak limited. As in the previous section, for disease diffusion model M and graph G, let $I_M(G)$ be the number of people that become infected due to a single infected individual, chosen uniformly at random, which spreads by M.

Restricted Disease problem (RD): Given a contact network $G = (V, E)$, budget B, and disease diffusion model M, find a minimum size subset $V' \subseteq V$ such that $E[I_M(G \setminus V')] \leq B$.

Since RD is a dual of Budgeted Vaccination, if we can solve either of these problems optimally then we can solve the other by a binary search over inputs to the other. More precisely, denote that the budget for Restricted Disease as B_{RD} and the denote the budget for Budgeted Vaccination as B_{BV}. To solve Budgeted Vaccination with budget B_{BV} suppose that we are given an oracle for finding the optimal solution to Restricted Disease. In polynomial time we could find, by considering all possible budgets, the minimum budget B_{RD} such that the solution produced by the oracle has cost no greater than B_{BV}. The solution returned by the oracle would then be an optimal solution to BV with budget B_{BV}. The same procedure can be used in the opposite direction, given an oracle for optimally solving Budgeted Vaccination, for solving instances of RD with budget B_{RD}.

However, since neither Restricted Disease nor Budgeted Vaccination can be solved optimally in general, we consider a number of “proxy” problems for Restricted Disease. These problems turn out to be NP-hard an thus we must rely on approximation algorithms or heuristics.

4.3.1 Partial Vertex Cover

If we ignore the complication introduced by the disease diffusion model M and suppose that we can bound the spread of disease by reducing the density of the contact network, then RD can be approximated by the *k-partial vertex cover* problem. That is, we think of wanting to “cover” a given number of edges as a way of limiting the spread of disease.

k-Partial Vertex Cover problem (k-PVC): Given a graph $G = (V, E)$ and positive integer k, find a minimum size set of vertices $V' \subseteq V$ that *covers* at least k edges in G. A vertex v is said to cover all the edges incident on it.

The k-PVC problem is a special case of the *k-partial set cover* problem [43]. The *k-partial set cover* problem is given a set of elements U and collection of subsets $\mathcal{S} \subseteq 2^U$ and it aims to find a minimum size set $S \subseteq \mathcal{S}$ that *covers* at least k members of U. An element $u \in U$ is said to be covered by $S \subseteq \mathcal{S}$ if there is a set $A \in S$ such that $u \in A$. In general this problem is NP-hard [44] but there are known approximation algorithms, utilizing a primal-dual approach, which provide an α-approximation given that no element $u \in U$ occurs in more than α sets in $\mathcal{S}$. The reduction from k-PVC is as follows. Each set $S_v \in \mathcal{S}$ corresponds to a vertex $v \in V$ such that S_v is the set of edges adjacent to v. Since each member u_e is in exactly two sets S_u, S_v, where u, v correspond to the endpoints of edge e, k-PVC can be approximated within a factor of 2 [43]. The algorithm for finding an α-approximate solution to k-PVC works by assuming the set A, for each set $A \in \mathcal{S}$, is in the solution and then solving the dual under this assumption. The algorithm and analysis of approximation for the α-approximate solution to k-PVC is given by Gandhi et al. [43] and relies on the LP given below.

The IP formulation of the partial vertex cover problem for a graph $G = (V, E)$ with n vertices and m edges and integer k is given as follows. For each vertex $v \in V$ we have a variable x_v and we say $x_v = 1$ iff S_v is in the solution S. Similarly we assign to each edge $e \in E$ a variable y_e and say that $y_e = 1$ iff edge e is "uncovered" by some set in S. The LP relaxation of this IP is,

$$\begin{aligned} & \min \sum_{v=1}^{n} x_v & \\ \text{subject to} \quad & y_e + \sum_{v:v\in e} x_v \geq 1 & e = 1, 2, \ldots, m \\ & \sum_{e}^{m} y_e \leq n - k & \\ & x_v \geq 0 & v = 1, 2, \ldots, n \\ & y_e \geq 0 & e = 1, 2, \ldots, m \end{aligned}$$

The first constraint makes sure that if an edge $e = (u, v)$ is covered, $y_e = 0$, then either x_u or x_v is greater than one, i.e., one of the edges endpoints is added to the solution set S. The second constraint makes sure that there are no more than $n-k$ edges that are uncovered in S.

The corresponding dual LP contains a variable v_e for each of the first m constraints and a variable z for constraint $\sum_e^m y_e \leq n-k$.

$$\max \sum_{i=e}^{m} v_e - (n-k)z$$

$$\begin{array}{llr} \text{subject to} & \displaystyle\sum_{e:e\in S_v} u_e \leq 1 & v \in V \\ & u_e \leq z & e = 1, 2, \ldots, m \\ & u_e \geq 0 & e = 1, 2, \ldots, m \\ & z \geq 0 & \end{array}$$

Experiments using the algorithm given by Gandhi et al. [43] shows that it performs much better than the worst case ratio on the HCW contact networks we generate. In Figure 4.6 we compare the approximation against a lower bound, found by solving the LP given above for k-PVC, and rounding it up. We round the fractional LP solution up because in practice solutions cannot be fractional.

From the Figure 4.6 we can see that for all values of k we tested, the algorithm actually finds solutions which approximate the optimal solution of k-PVC a factor of at most 1.25. For small values of k the solutions returned by the approximation are optimal. While we want to cover as many edges as possible properly vaccinating the "right" individuals may not require that all edges need to be covered. And so while we would like the approximation ratio to be optimal, we may also not have to worry about large values of k.

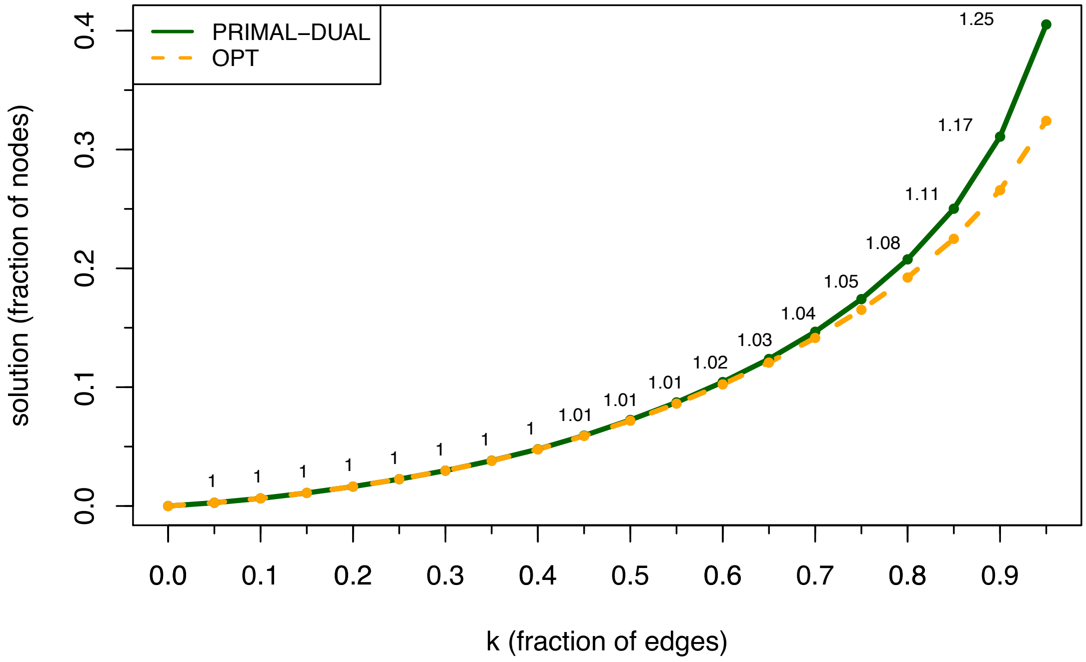


Figure 4.6: Experimental comparison of the solution to k-PVC produced by Gandhi et al. [43] with the lower bound found by solving the corresponding LP. Points are annotated with the approximation ratio relative to the lower bound.

Interestingly, solutions to k-PVC are solutions to certain instances of the Max-BVC. k-PVC is given as input a graph G and integer k. The solution returned by k-BVC is a minimum size set of vertices V' and covers at least k edges. If we denote the size of V' as I, then V' is also a solution to the Max-BVC problem with budget of vertices $B = l$; removing the set V' from G will remove at least k edges. Thus, we can find a solution to an instance of BVC $\{G, B\}$ by repeatedly "guessing" values of k and finding a solution V' to the k-PVC instance $\{G, k\}$, until we find maximum k such that $|V'| \leq B$. Since k is bounded by the number of edges in G, m, we can find a solution in $O(\log m)$ time. However, there is no guarantee on how good of a solution V' is relative to OPT for the corresponding instance of BVC.

Experimental runs comparing greedy solutions for BVC and k-PVC (given in Figure 4.7) show that both solutions are nearly equivalent. That is, solutions obtained by the primal-dual approximation for k-PVC are as good as the approximations obtained by the greedy for corresponding instances of BVC, and vice versa. Since the **GreedyBVC** is far easier to solve, this suggests that using BVC to approximate RD is as good as approximations obtained by solving k-PVC.

4.3.2 Restricted Max Degree

Recall that in the BMD problem we place a bound on the number of vertices to remove and the objective is to minimize the maximum degree. A "dual" formulation of BMD would be one in which we want to bring the maximum degree below some given threshold while removing the fewest number of vertices. We call this problem

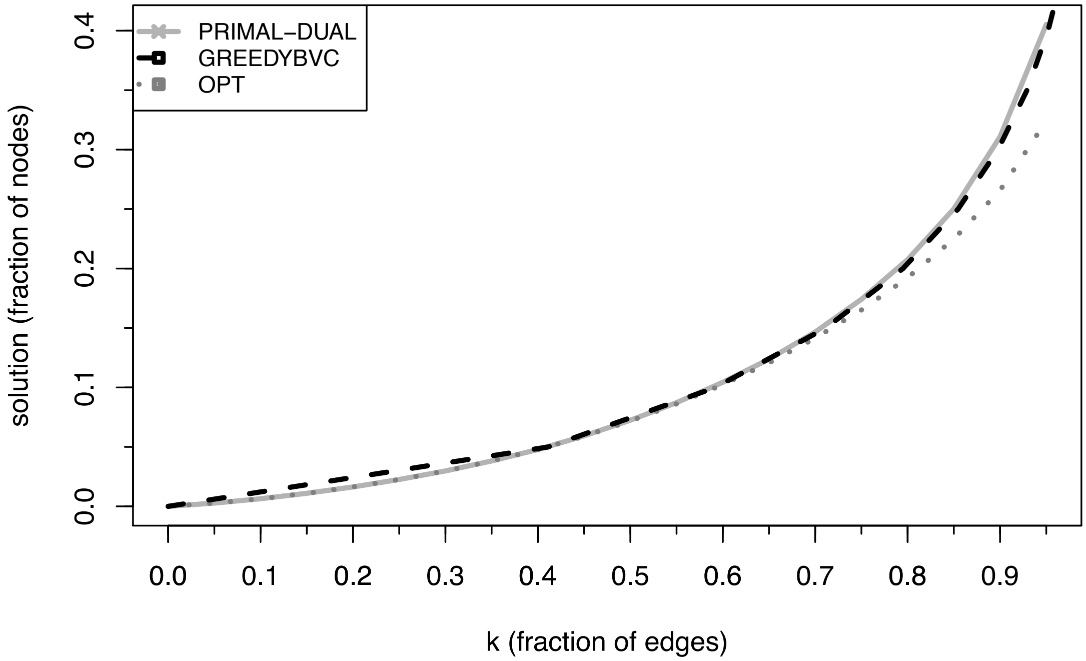


Figure 4.7: Comparison of approximate solutions for k-PVC used to solve instances of Max-BVC. The dashed line shows the result from **GreedyBVC** , the solid line shows the solution found by the primal-dual k-PVC algorithm given by Gandhi et al. [43], and the dotted line shows the lower-bound of the k-PVC problem based on the LP solution.

Restricted Max Degree problem.

Restricted Max Degree (RMD): Given a graph $G = (V, E)$ and maximum degree threshold k, find a minimum size subset $V' \subseteq V$ such that,

$$\max_{v \in V \setminus V'} d'(v) \leq k$$

where $d'(v)$ is the degree of vertex v in the graph $G' = G \setminus V$ resulting from removing V'.

RMD can be reduced to a well-known generalization of set cover called *multiset multicover*. In the *multiset multicover* problem we are given a collection of multisets $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ over a set U and collection of demands $R = \{r_u | u \in U\}$ for each element in U. Let $M(S, e)$ denote the multiplicity of e in the multiset $S \subseteq U$. The objective of *multiset multicover* is to find a minimum size subset $S \subseteq \mathcal{S}$ such that

$$M(\bigcup_{S_i \in S} S_i, u) \geq r_u$$

for each element $u \in U$.

Theorem 4.3. *If there is a β-approximation for MSMC then there is a β-approximation for RMD.*

Proof. Let $H = (G, k)$ be an instance of RMD where $N(v)$ denotes be the set of neighbors of v and $d(v) = |N(v)|$ denotes the degree of vertex v. To construct an instance $H' = (U, \mathcal{S}, R)$ of *multiset multicover*, let $U = \{v \in V | d(v) > k\}$ and $R = \{r_v | v \in U\}$ where $r_v = d(v) - k$. To construct $\mathcal{S}$ we construct a $S_v \in \mathcal{S}$ for each $v \in V$ in the following manner. For vertex v we add $d(v) - k$ copies of v, such that $M(S_v, v) = d(v) - k$ and a single instance of every vertex $u \in N(v)$ such that $M(S_v, u) = 1$. Thus each multiset S_v is of size $2d(v) - k$.

Suppose that is an algorithm A that gives a β approximation to multiset multicover. Starting with an instance $H = (G, k)$ we reduce to an instance $H' = (U, \mathcal{S}, R)$ as outlined above. Running A on H' will return a subset $S \subseteq \mathcal{S}$ such that $|S| \leq \beta|OPT|$ where $OPT \subseteq \mathcal{S}$ is the optimal solution to the instance H'. Each multiset $S_u \in S$ corresponds to a vertex in the graph G such that u's removal from the graph will decrease it's own degree to 0 and will decrease the degree of its neighbors by 1. And by the definition of the MSMC problem, the solution S covers every element $u \in U$ at least r_u times. Note that each element in u corresponds to an vertex in the graph G with degree $d(u) > k$ and the requirement $r_u = d(u) - k$. Let $V' \subseteq V$ be the subset of vertices v such that $S_v \in S$. Then removing V' from G will reduce the degree of vertex u to at most $d(u)$. Thus S is a feasible solution to H. By the same argument the set of subset of vertices $V_{OPT} \subseteq V$, corresponding to each v such that $S_v \in OPT$, is an optimal solution to the RMD problem. And since $|S| \leq \beta|OPT|$ then $|V'| \leq \beta|V_{OPT}|$.

The best known approximation for the *multiset multicover* is a simple greedy algorithm that repeatedly picks the multiset that satisfies the most element requirements. This greedy algorithm provides an $O(\log(m))$ approximation guarantee where m is size of the largest multiset. By Theorem 4.3 the greedy algorithm for MSMC provides an $O(\log(z))$ approximation for RMD where $z \leq 2\Delta(G) - k$ and $\Delta(G)$ is the maximum degree of any vertex in G. The bound on z is due to the reduction to

MSMC. Notice that the maximum degree vertex in G, say v', will have $d(v') - k$ copies of itself and a single copy of each of its neighbors $u \in N(v')$ in the corresponding $S_{v'}$ given by the reduction to MSMC. And thus the largest multiset after the reduction will have size $2\Delta(G) - k$.

Like the other optimization problems considered before, we can formulate RMD as an LP in the following way. Let $G = (V, E)$ be a graph with n vertices and m edges, and let k be an integer. For each vertex $v \in V$ assign a variable x_v and say $x_v = 1$ iff $v \in S$ (the solution set for RMD). For each edge $e \in E$ assign a variable y_e such that $y_e = 1$ iff e is "uncovered" by a vertex in the solution set S. Then the LP (already a relaxed from the IP) is,

$$\min \sum_{j=1}^{n} x_j$$

$$\begin{array}{llr} \text{subject to} & y_i + \sum\limits_{j:j \in e_i} x_j \geq 1 & i = 1, 2, \ldots, m \\ & \sum\limits_{i:n_j \in e_i} y_i \leq k & j = 1, 2, \ldots, n \end{array}$$

We solve the LP of RMD and compare it to experimental runs of the greedy algorithm for MSMC. As before, we round up the LP solution since valid solutions to RMD are not fractional. Results are shown in Figure 4.8.

4.3.3 Discussion

In general, optimal solutions considered by researchers as "proxy" problems do not provide optimal solutions to BV and RD. Since these "proxy" problem are

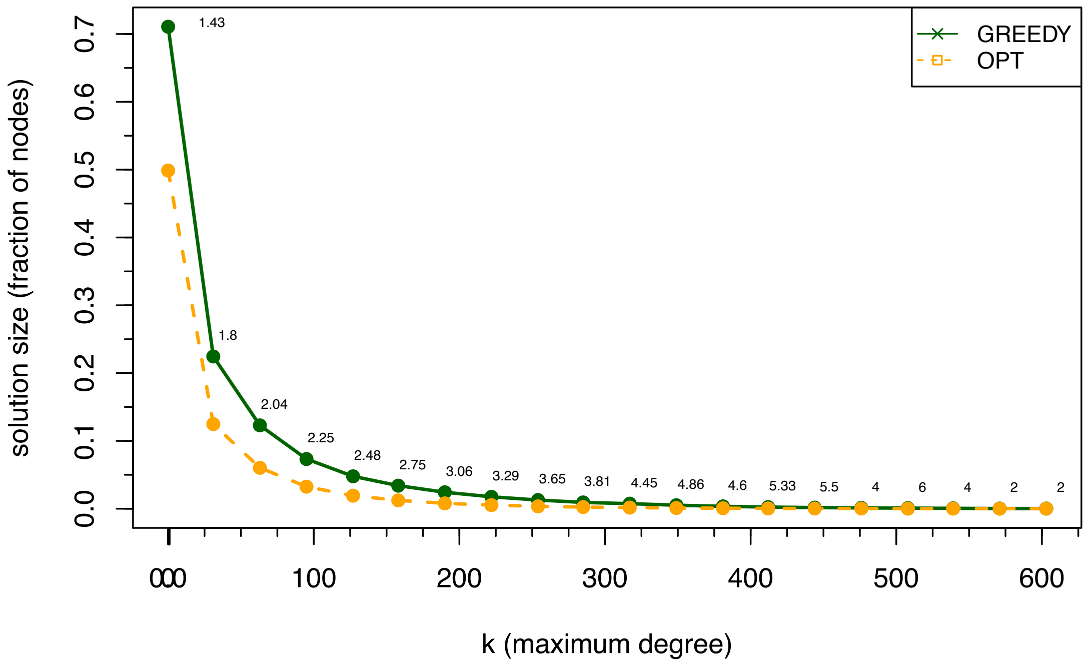


Figure 4.8: Comparison of experimental runs of the greedy solution to RMD an upper bound calculated by solving the LP relaxation on the $moderate_1$ graph. LP solutions are rounded up to the nearest integer. The x-axis denotes the integer k and the y-axis denotes the solutions size give as a fraction of the total number of vertices.

themselves NP-hard, and thus we can only achieve approximate solutions, makes the situation is even worse. However, we have shown significant evidence that suggests in most cases, simply solving Max-BVC, utilizing **GreedyBVC**, provides good solutions to BV and RD on our HCW contact networks.

4.4 Experimental Analysis of Simple Vaccination Policies

Using generated HCW contact graphs as a proxy for disease-spreading contacts within the UIHC population, we compare the effectiveness of several different vaccination policies. We assume that any vaccination that is administered is 100% effective and effective immediately. This assumption allows us to model the action of vaccinating a person v as the deletion of the vertex v from the HCW contact graph. Given a budget of vaccinations b, a vaccination policy tells us which b people from the population to vaccinate. More precisely, given a budget $b \geq 0$, a *vaccination policy* is a probability distribution over all size-b subsets of the population indicating the likelihood of a particular size-b subset being chosen. To a large extent our work focuses on the special case of vaccination policies that are deterministic, i.e., ones that assign probability 1 to exactly one size-b subset and probability 0 to all others. Here we evaluate five simple vaccination policies; *random*, *degree-based*, *weighted-degree-based*, *distance-based*, and *computers-based.*

Random policy. Repeatedly pick an individual for vaccination by sampling vertices uniformly at random from the HCW contact network.

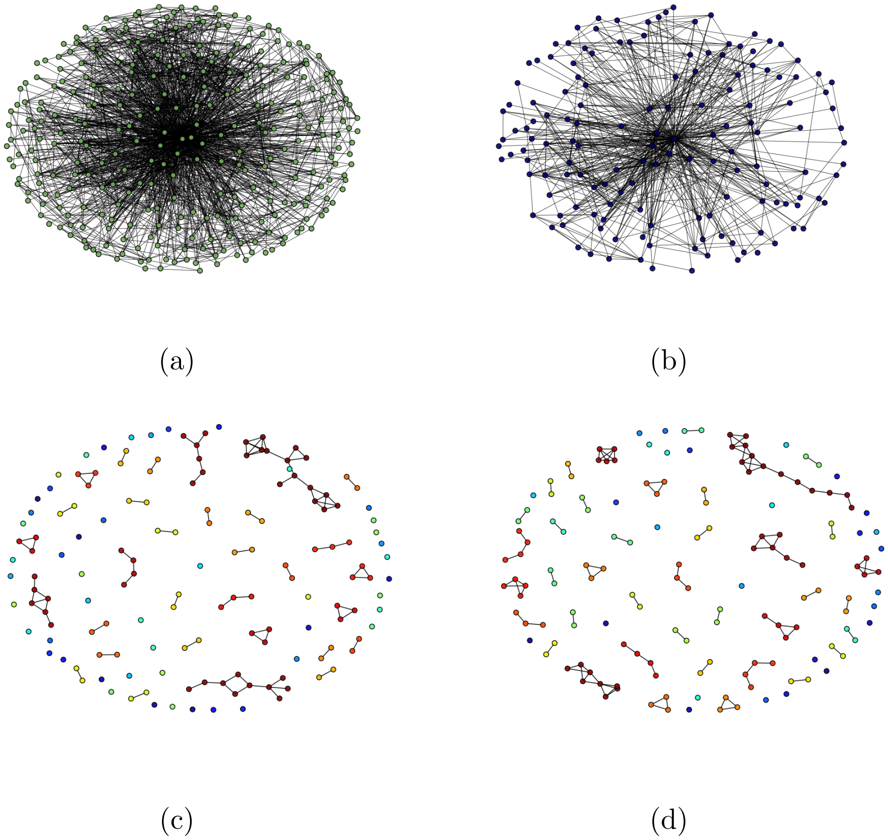
Degree-based policy. Let $G = (V, E)$ be a HCW contact network and let $S \subseteq V$ be the set of HCWs already vaccinated. Repeatedly pick for vaccination a person with highest degree in $G - S$. Break ties uniformly at random.

Weighted-degree-based policy. Let $G = (V, E)$ be a HCW contact network. Each edge $\{u, v\} \in E$ has an associated weight $w(u, v)$ that represents the the number of contacts between individuals u and v during time window T. Define the *weighted degree* of a vertex v in a HCW contact network G as $\sum_{u \in N(v)} w(u, v)$, where $N(v)$ is the set of neighbors of vertex v in G. Let $S \subseteq V$ be the set of already vaccinated healthcare workers. Repeatedly pick for vaccination a person with highest weighted degree in $G - S$. Break ties uniformly at random.

Distance-based policy. Let the *distance traveled* by a HCW in a time window T be the sum of shortest-path hop distances between the locations of consecutive logins within that window. Repeatedly pick the person with the most distance traveled during time window T. Break ties uniformly at random.

Login-heterogeneity-based policy. Repeatedly pick the healthcare worker who logs into the most distinct computers in time window T. This is given by the individual's degree in the computers-people graph for time window T (see Chapter 2 for discussion on the computers-people graph). Break ties uniformly at random.

The random policy is oblivious to the characteristics of individual HCWs, picking uniformly at random from the population. The degree-based and weighted-degree-based policies pay attention "connectivity" characteristics. The distance-based and login-heterogeneity-based policies pay attention to "spatial" characteristics.



(a)

(b)

(c)

(d)

Figure 4.9: Example vaccination policies on a subgraph of an EMR contact network. (a) Small portion of the $sparse_1$ contact network (b) Results of vaccinating 50% of the population using the random policy. The result is still one single component. (c) Results of vaccinating 50% of the population using the degree-based policy. (d) Results of vaccinating 50% of the population using the distance-based policy. In both (c) and (d) the HCW contact network is "shattered" into many tiny components.

The effectiveness of a vaccination policy is, in a sense, how well implementation of the policy reduces disease spread. Let G be a HCW contact network and let G' be the graph that is obtained from G by deleting the healthcare workers selected by the policy. Figure 4.9 illustrates how different policies can affect different outcomes. In this example the degree-based and distance-based "shatter" the graph, separating vertices into small connected components, versus the random policy which results in a single connected component with decreased density.

To experimentally measure the effectiveness of a vaccination policy we use two metrics.

1. The expected number of infected people in G' based on an SIR model M. Specifically, we use the SIR model M as defined in Section 4.1 and measure $E[I_M(G')$. Since efficiently calculating $E[I_M(G')]$ directly is an open problem [54] we take average from repeated runs of agent-based simulations. We use a method called *fast-diffuse* [108] that allows each simulation in time roughly linear in the number of edges. For this model, 100 simulations take approximately 3 seconds and we run $10,000$ simulations per graph G'.

2. Expected size of the largest connected component in G'. This is a deterministic worst-case measure that supposes that any individual who comes in contact with an infected individual, also becomes infected. This measure has the advantage of being computationally cheap, computed in time linear to the size of the graph, but it ignores disease characteristics and differences in strengths of contacts.

Figures 4.10 and 4.11 compare the five policies using agent-based simulations of a disease diffusion model, introduced in Section 3.4, which emulates the spread of influenza. In these experiments we characterize our simulations by the "peak transmission" probability p that specifies the probability of transmitting disease on the second day of infection. Recall that each edge (u, v) in our HCW contact networks have associated edge weight $w(u, v)$. More specific information on how edge weights are used these simulations is given in Section 3.4. An interesting aspect of these plots is that with the lower transmission probability $p = .07$ the weighted-degree policy does slightly better than the degree policy, whereas with higher transmission probability $p = .5$ the degree policy does better. This difference is likely due to the fact that when transmissibility is high, even low weight edges will have a high probability of transmitting disease. This follows our intuition that when disease has a low transmissibility, low weight edges are less likely to transmit disease and thus looking at the weight of edges, as opposed to the degree, is more important.

Figures 4.12, 4.13, and 4.14 compare the five vaccination policies on the $sparse_1$, $moderate_1$, and $dense_1$ graphs, respectively, using the size of the largest connected component. Recall that $sparse_i$, $moderate_i$, and $dense_i$ graphs correspond to the timewindow $T = i$ and graphs generated with parameters $d = 1, t = 0$, $d = 3, t = 15$, and $d = 5, t = 30$ respectively.

The first thing to note is that the connected-component measure preserves the relative differences between policies when transmissibility is high. But since the connected component measure is irrespective of transmissibility, it is unable to identify

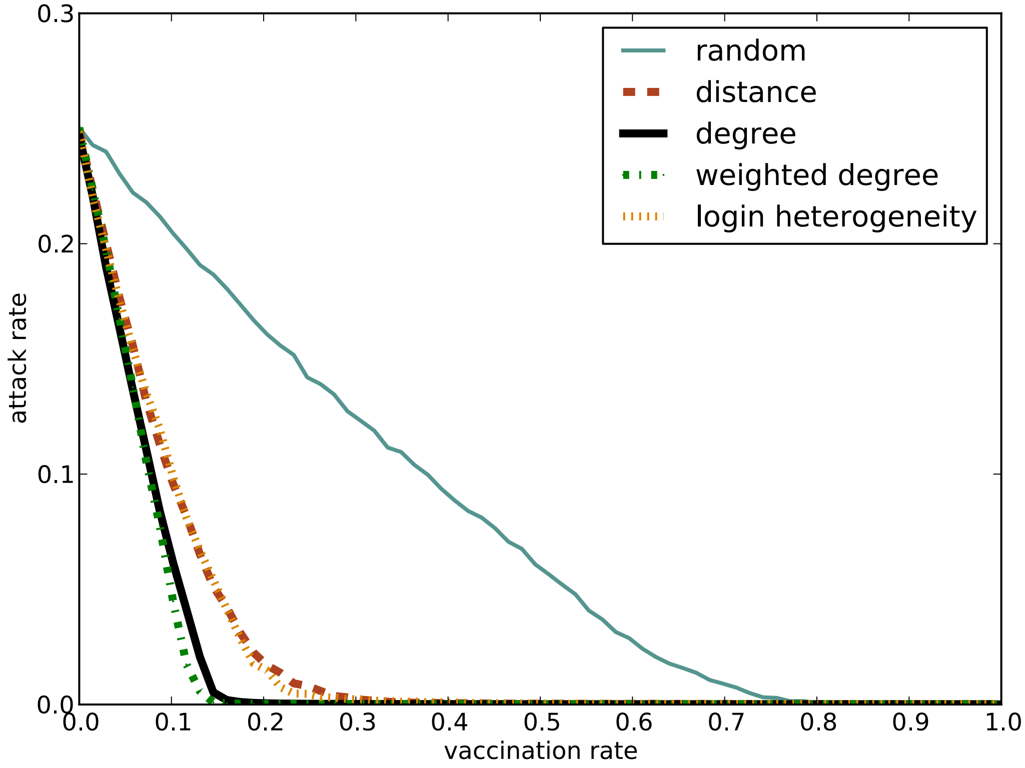


Figure 4.10: Effectiveness of vaccination policies on the $moderate_1$ HCW contact networks measured by the expected number of people infected as a result of simulation of an SIR disease diffusion simulation with peak transmission probability $p = .07$. The x-axis represents the budget of vaccinations as a fraction of the total population. The y-axis represents the expected fraction of the population infected based on experiments.

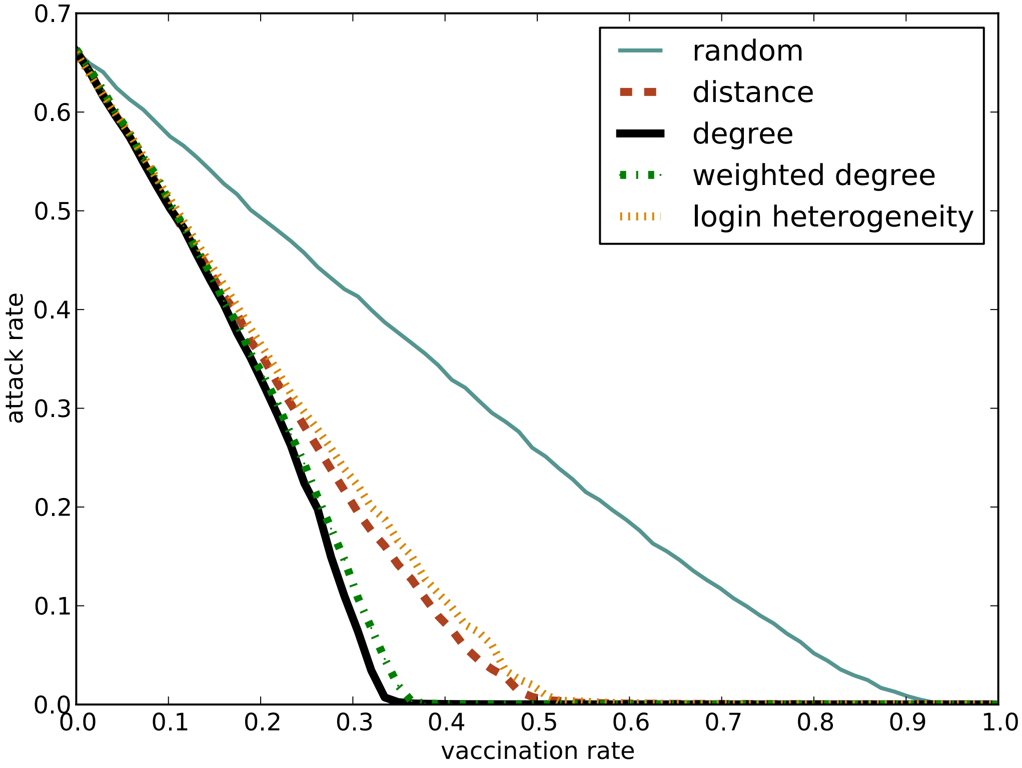


Figure 4.11: Effectiveness of vaccination policies on the *moderate*$_1$ HCW contact networks measured by the expected number of people infected as a result of simulation of an SIR disease diffusion simulation with peak transmission probability $p = .5$. The x-axis represents the budget of vaccinations as a fraction of the total population. The y-axis represents the expected fraction of the population infected based on experiments.

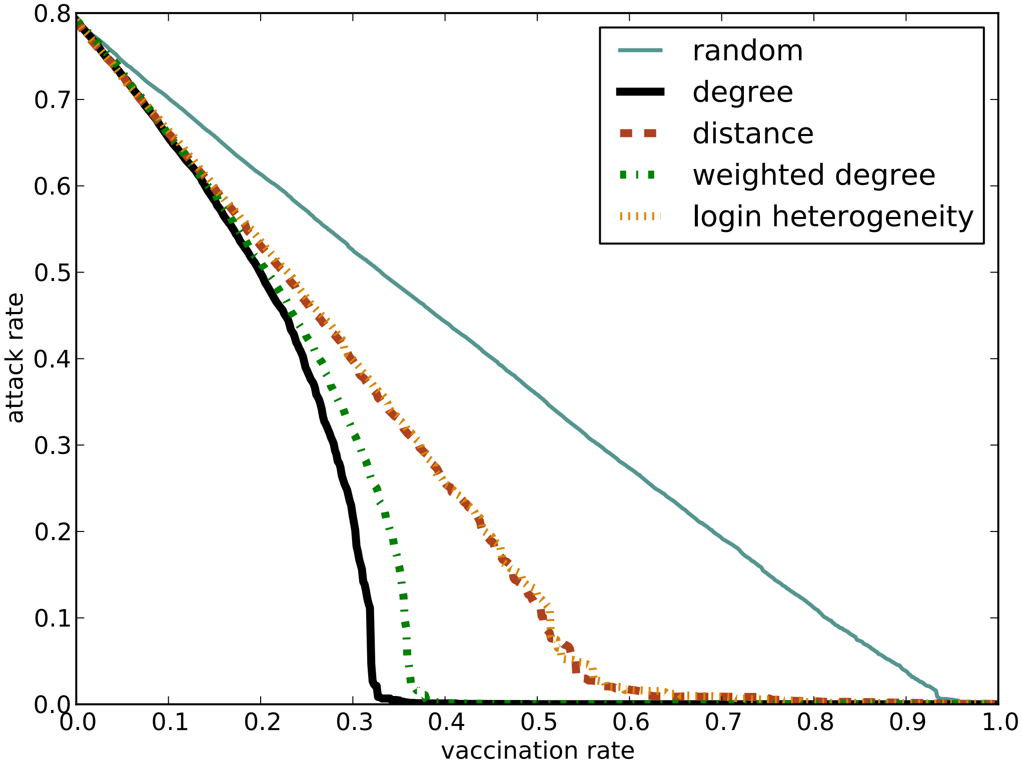


Figure 4.12: Effectiveness of five different vaccination policies on the $sparse_1$ contact network as measured by the size of the largest connected component in the unvaccinated network. The x-axis represents the percentage of people vaccinated. The y-axis represents the size of the largest connected component as a percentage of the size of the entire graph.

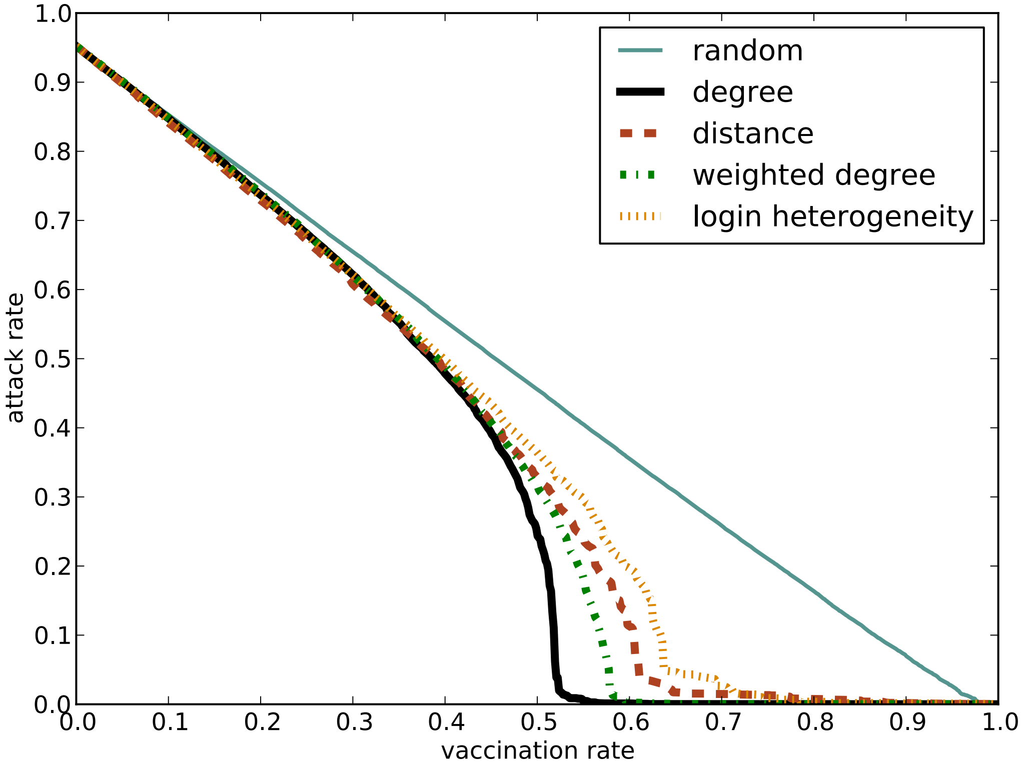


Figure 4.13: Effectiveness of five different vaccination policies on the $moderate_1$ contact network as measured by the size of the largest connected component in the unvaccinated network. The x-axis represents the percentage of people vaccinated. The y-axis represents the size of the largest connected component as a percentage of the size of the entire graph.

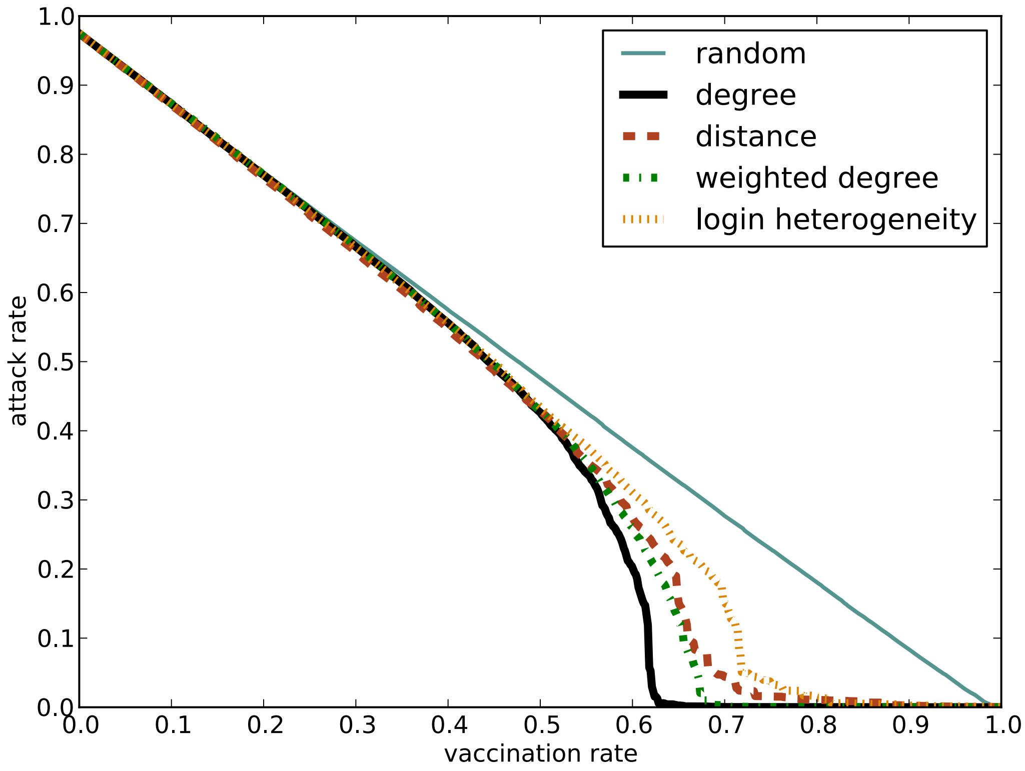


Figure 4.14: Effectiveness of five different vaccination policies on the $dense_1$ contact network as measured by the size of the largest connected component in the unvaccinated network. The x-axis represents the percentage of people vaccinated. The y-axis represents the size of the largest connected component as a percentage of the size of the entire graph.

the advantage that the weighted-degree policy has over the degree-based policy in Figure 4.13. This relates back to our earlier discovery that policies which assume a high level of transmissibility may fail to adequately capture real-world behavior when transmissibility is low. That is, measuring these policies without concern for transmissibility can be inaccurate in some cases.

In all of our experiments the connectivity-based policies, degree-based and weighted-degree-based, perform consistently better than other policies. Still, the spatial-based policies, distance-based and login-heterogeneity-based, perform quite well relative to the random policy. Figure 4.15 gives a scatter plot showing the correlation between degree and distance for each vertex in the *moderate*$_1$ graph. There appears to be minor correlation between degree and distance but there are a number of outliers, specifically vertices with high distance traveled and low degree, that are likely causing the policy differences we see. One advantage to the spatial-based policies, in practice, is that they are easier to implement. We can imagine that asking HCWs to wear pedometers or a simple analysis of login records would be feasible for hospital administrators to implement in order to inform a targeted vaccination campaign.

The experiments presented thus far give the connectivity-based policies an unfair advantage by evaluating them on the very same networks that they were generated from. A more realistic evaluation would generate connectivity-based policies on a particular HCW contact network, but evaluate these on a different, but structurally similar network. Since we have many HCW contact networks at our disposal, such an

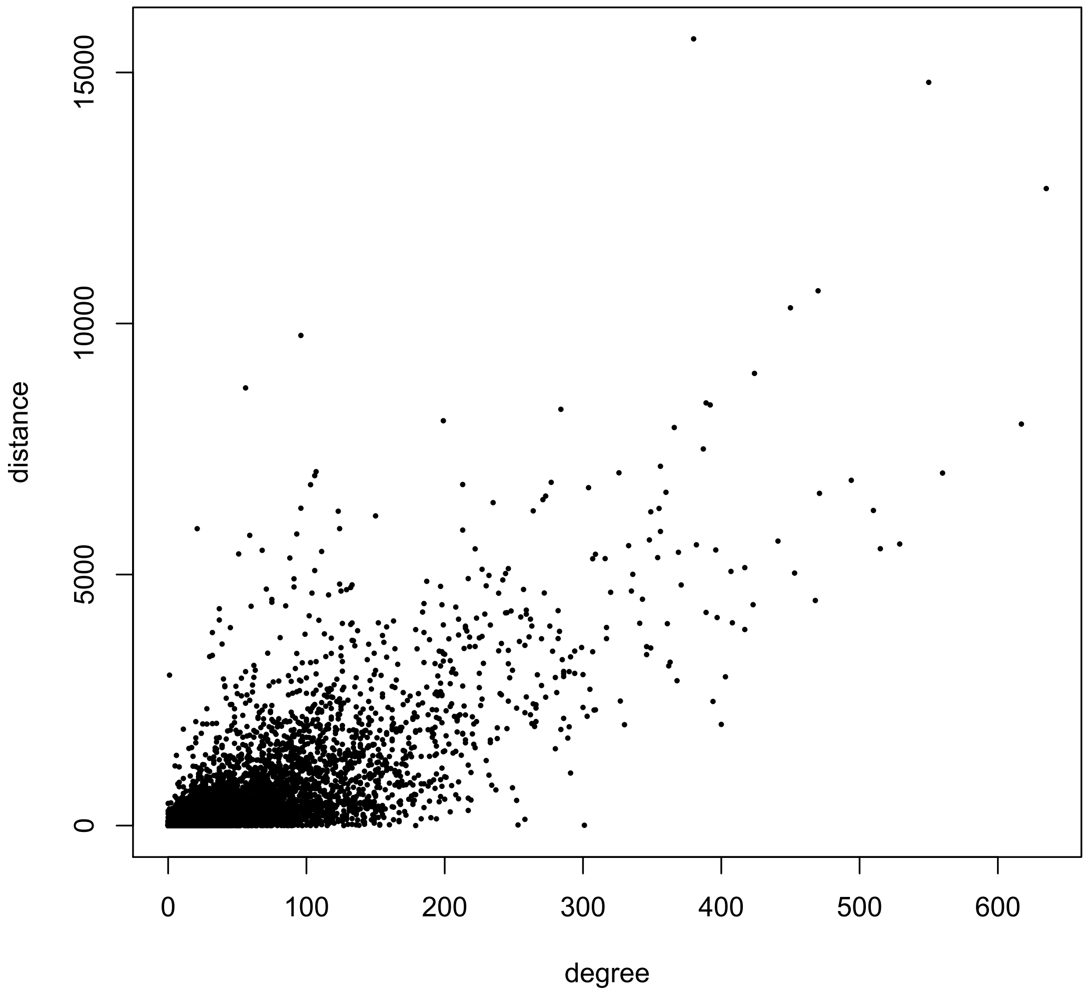


Figure 4.15: Comparison of distance traveled, as measured by number of hops between consecutive logins, versus the degree of each vertex in the $moderate_1$ graph.

evaluation is easy and is shown in Figure 4.16. Even though the connectivity-based policies still outperform the mobility-based policies, the difference is much less striking. But also remember that the HCW contact networks are simply an approximation of actual interactions of HCWs for a single time slice and as these actual networks are constantly evolving. From our limited experiments it appears that distance is more resilient to this change in the contact network.

To validate the behavior of our vaccination policies is consistent over all time periods that we have login data for, we plot effectiveness as measured by the size of the expected number of infected individuals for eight different four-week periods spread out throughout the entire 22 month period for the degree and distance policies. Validation is shown in Figure 4.17. While between time periods there is some minor variation, likely due to minor differences in the network structure, the relative behavior between different policies is consistent between time windows.

For completeness we have also considered solutions to k-PVC and RMD as possible vaccination policies. Recall from Section 4.3.1 that the k-PVC is to find a minimum size set of vertices that covers at least k edges. The sets of vertices that is returned as solutions to the k-PVC can be considered as vaccination policies. Figure 4.18 shows experimental runs of the k-PVC-based policy compared to the degree and distance based policies. As we can see polices based on k-PVC-based do much better than the spatial-based policies, but there appears to be no advantage to considering the k-PVC-based policy over the degree policy. It is not entirely surprising that policies derived from solutions to k-PVC do not outperform the degree policy.

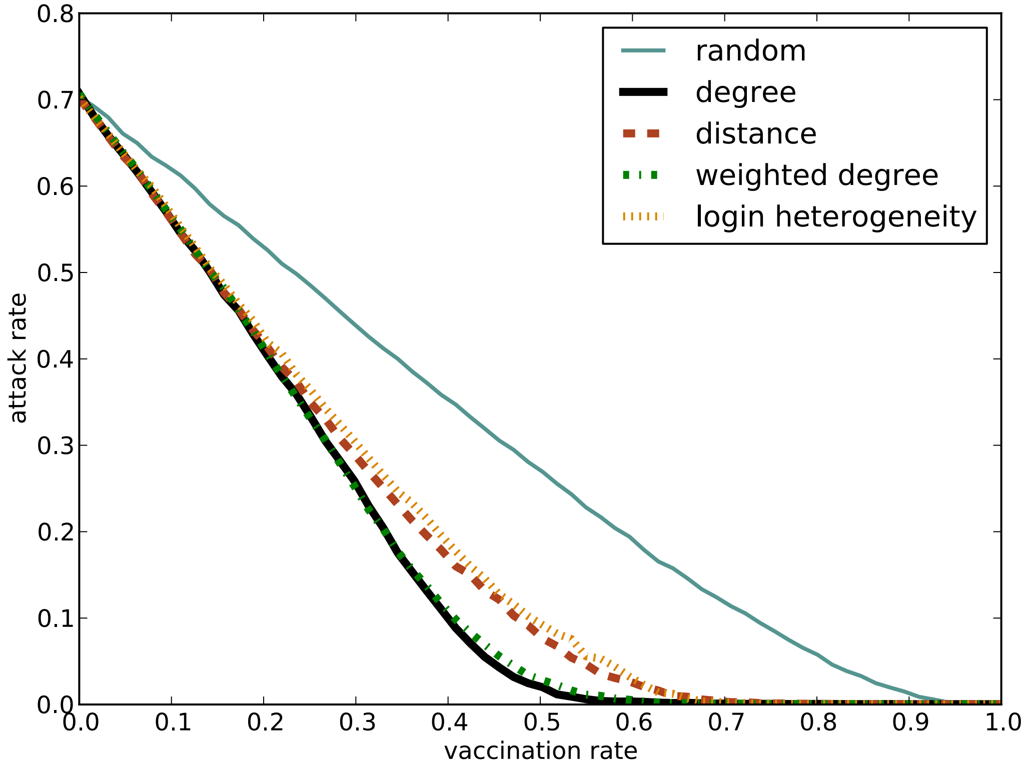
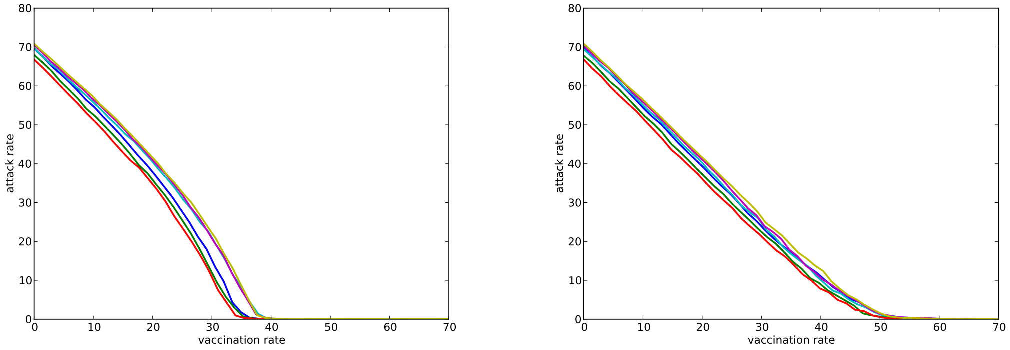


Figure 4.16: Effectiveness of vaccination policies on a "time-shifted" HCW contact network. The connectivity-based policies are generated from the *moderate*$_1$ HCW contact network. The mobility-based vaccination policies are generated using the EMR login data for the same time window $T = 1$. The plot shows the effectiveness of these policies on the "time-shifted" HCW contact network, *moderate*$_5$, measured by the expected number of people infected starting from a single infected individual chosen uniformly at random. The $T = 1$-network and the $T = 5$-network not only differ in edges, but also in the HCWs they contain as vertices.



(a)

(b)

Figure 4.17: Vaccination policies (a) degree and (b) distance for eight different *moderate* contact networks generated from four week periods spread throughout the 22 month period we have data for. Polices are evaluated by measuring the size of the largest connected component after vaccination. The x-axis represents the budget of vaccinations as a fraction of the total population. The y-axis represents the expected fraction of the population infected based on experiments.

Note that the degree-based policy is the solution to BVC returned by the **GeedyBVC** algorithm. BVC is, roughly speaking, to minimize the number of edges given a budget of vertices, and **GreedyBVC** provides a $(1-1/e)$ approximation. On the other hand, k-PVC is the "dual" problem to BVC, but the approximation used to solve k-PVC is a 2-approximation and makes no guarantee about the quality of the solution return as applied to BVC.

Similarly we can consider solutions to RMD as possible vaccination policies. Recall from Section 4.3.2 that RMD is to find a minimum size set of vertices whose removal from the graph will lower the maximum degree below some threshold k. From Figure 4.19 we can see that, like the k-PVC-based policy, the RMD-based policy does much better than the distance-based policy but does not appears to provide an

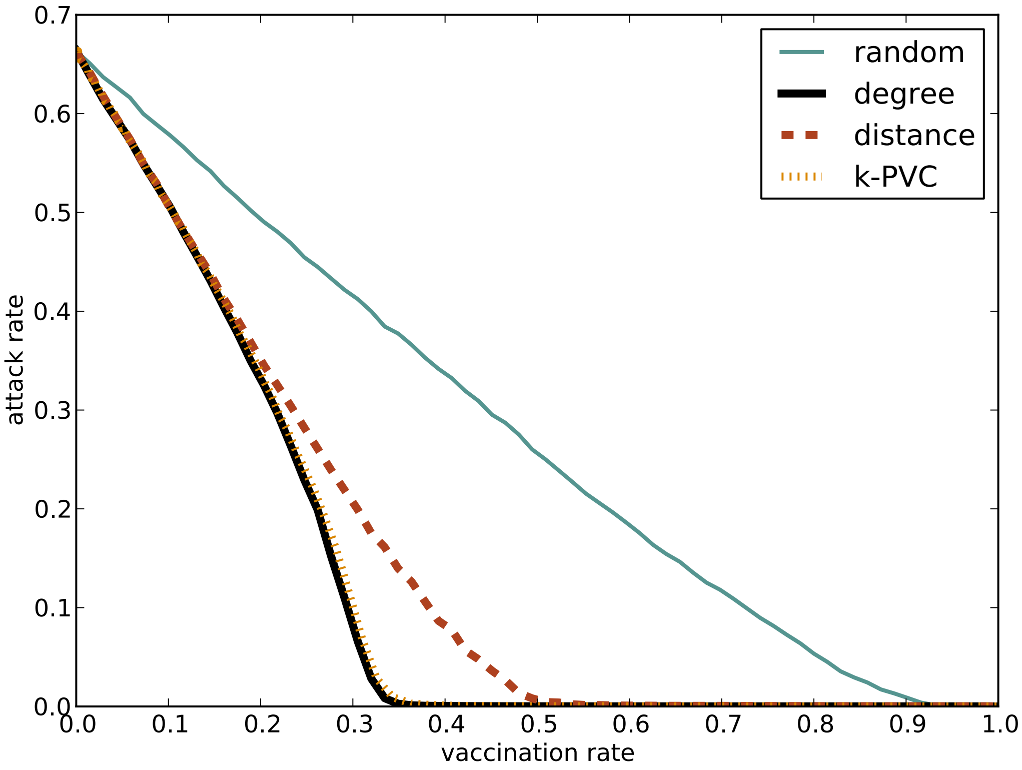


Figure 4.18: Effectiveness of the degree-based, distance-based, k-PVC-based, and anonymous policies on the *moderate*$_1$ HCW contact networks measured by the expected number of people infected as a result of simulation of an SIR disease diffusion simulation with peak transmission probability $p = .5$. The x-axis represents the budget of vaccinations as a fraction of the total population. The y-axis represents the expected fraction of the population infected based on experiments.

advantage over the degree-based policies.

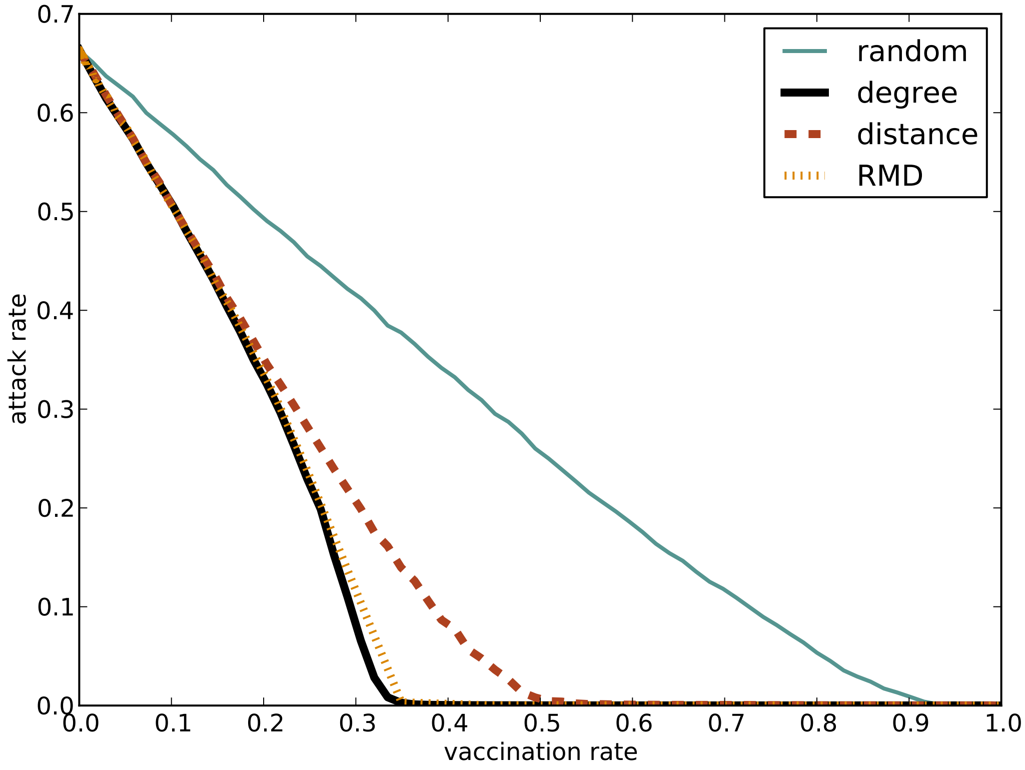


Figure 4.19: Effectiveness of the degree-based, distance-based, RMD-based, and anonymous policies on the *moderate*$_1$ HCW contact networks measured by the expected number of people infected as a result of simulation of an SIR disease diffusion simulation with peak transmission probability $p = .5$. The x-axis represents the budget of vaccinations as a fraction of the total population. The y-axis represents the expected fraction of the population infected based on experiments.

4.4.1 Improving on Simple Heuristics

To improve on the degree-based policy, we consider an enhancement that takes into account the "community structure" of the underlying network. Roughly speaking, our idea is to identify *cuts* with very low transmissibility. A cut is a subset of edges of the network whose removal increase the number of connected components by at least one. Suppose that $G = (V, E)$ is a connected graph and C is a cut in G such that $G \setminus C$ has two connected components G_1 and G_2. If C has very low transmissibility, then a disease that is initiated in G_1 is unlikely to infect any individual in G_2. This implies that vaccination policies can ignore the edges in C.

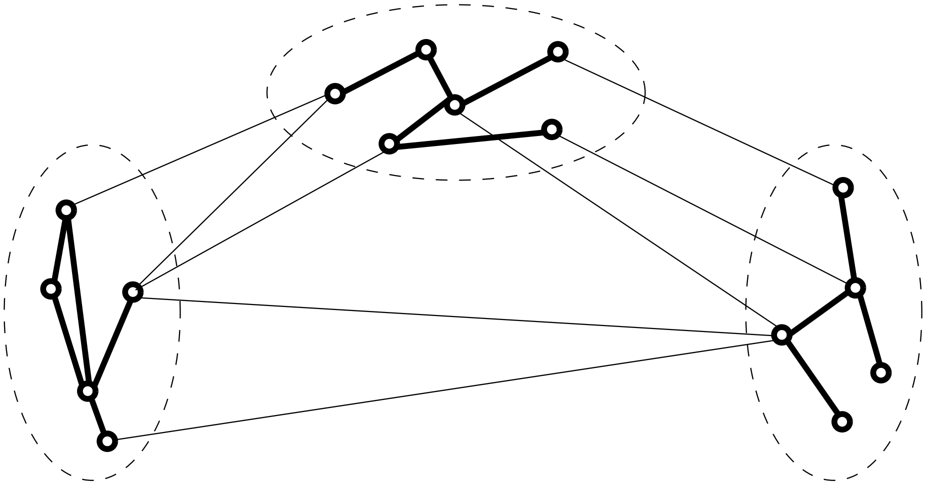


Figure 4.20: Example graph of communities with high weight intra-community edges and low weight edges between communities. Thin lines represent low-weight edges and thick lines represent high-weight edges. Communities are circled by a dotted line.

Based on this idea we introduce a policy called the *mincut-degree* policy described as follows. Suppose that we are given HCW contact network $G = (V, E)$ for time period T (in days), a fractional threshold *thresh*, and peak transmission probability p. As in the previous chapters, p denotes a vector of probabilities $\{p_1, \ldots, p_k\}$ where p_i denote the disease transmission probability for day i. For edge e let $tr(e) =$

$1 - \prod_j^k (1 - p_j)^{w'(e)}$ be the probability of disease transmission across edge e, where $w'(e) = w(e)/T$ is the weight of edge e averaged over the time period T, and for set of edges X let $tr(X) = \sum_{e \in X} tr(e)$. Further, denote the connected components of graph H as *components*(H) and let *mincut*(H) be the set of edges in the minimum edge cut on H. The first step in the *mincut-degree* policy is to find a set of non-influential edges E' (i.e., edges that are unlikely on their own or when combined with other edges to be vehicles for disease spread).

Algorithm 4.1 First step of *mincut* policy

```
$C \leftarrow components(G)$
$E' \leftarrow \emptyset$
while $C \neq \emptyset$
    Pick $H$ from $C$ uniformly at random.
    $C \leftarrow C \setminus \{H\}$
    $M \leftarrow mincut(H)$
    if $tr(M) \leq thresh$
        $C \leftarrow C \cup components(H - M)$
        $E' \leftarrow M$
    end if
end while
return $E'$
```

Roughly speaking, Algorithm 4.1 repeatedly calls the mincut algorithm on the connected components of G and either removes the edges in the minimum cut, in the case that the probability of disease transmission across the cut is no greater than *thresh*, or ignores the component as a candidate for future cuts, in the case that disease transmission across the cut is greater than *thresh*. The result of this stage is a set of edges E' that fall across cuts that have low transmission probability (i.e., less than *thresh*). For our experiments we rely on the min-cut algorithm included in the igraph library [30] which is based on the min-cut algorithm by Stoer and Wagner [104].

The second part of the *mincut-degree* policy is to perform degree policy on the graph $G' = (V, E \setminus E')$ where E' is the output from running algorithm 4.1 on G. Removing edges E' effectively ignores edges which will be unlikely to transmit disease and allows the degree policy to focus on vertices with high degree of "relevant" edges.

Even though in theory the mincut-degree policy should perform better than the degree policy, in practice, runs of the *mincut-degree* policy reveal no significant improvement over the degree-based policy (see Figure 4.21).

Table 4.1: Parameters used in testing the mincut-degree-based policy.

p	$\{.01, .02, \ldots, .09, .1, .2, \ldots, .6\}$
thresh	$\{.01, .02, .03, .04, .1, .2, .3, .4, .5\}$

We examine the graphs resulting from removing the edges returned by the process defined in Algorithm 4.1 for all pairs of p and *thresh* given in Table 4.1. After removing edges the connected component distribution remains one giant component and a large number of tiny components, mostly singleton vertices. This suggests mincut algorithm is "trimming" vertices from the boundary of the graph as illustrated in Figure 4.24. This behavior is likely the consequence of the positive assortativity

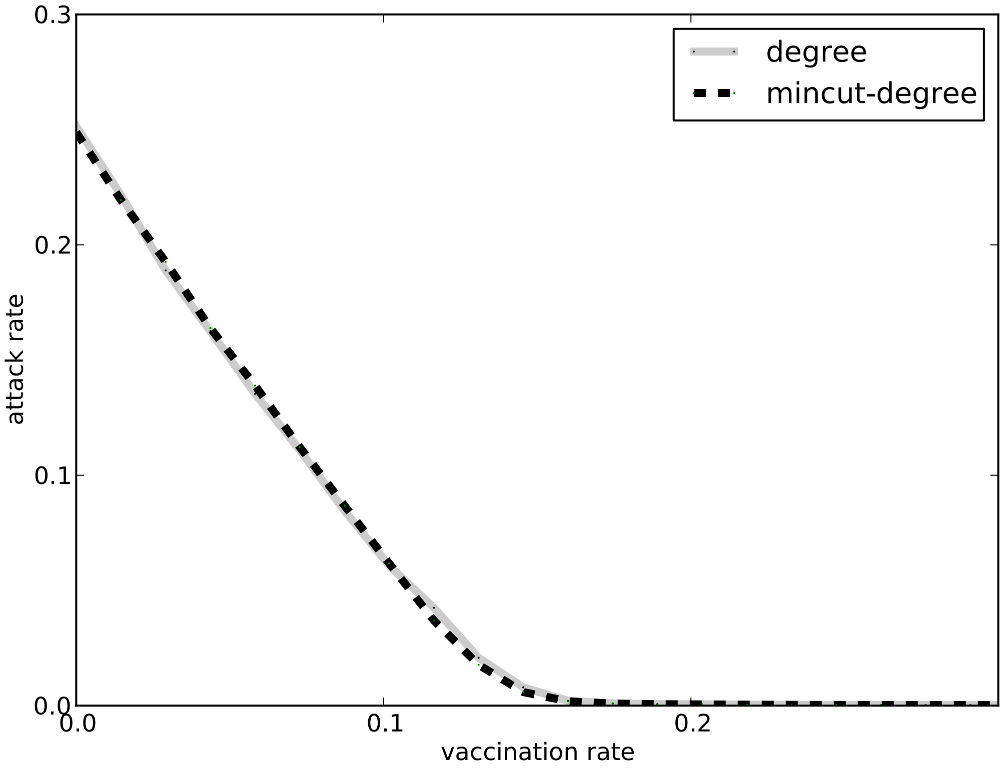


Figure 4.21: Comparison of the effectiveness of the mincut-degree-based policy and degree-based policy. Policies are generate from the *moderate*$_1$ graph for time window $T = 1$, peak transmission probability $p = .07$, and $thresh = .3$

Table 4.2: Statistics for the graph resulting from removing edges returned by the mincut process in Algorithm 4.1.

	$p = .07$ $thresh = .3$	$p = .1$ $thresh = .3$	$p = .5$ $thresh = .5$
Edges Removed	16,508	10,216	2,527
Giant Comp. Size	4414	4816	5698
Second Largest Comp. Size	8	8	6

Gives the size of the giant component, size of the second largest component, and the number of edges removed from the graph due to the mincut algorithm.

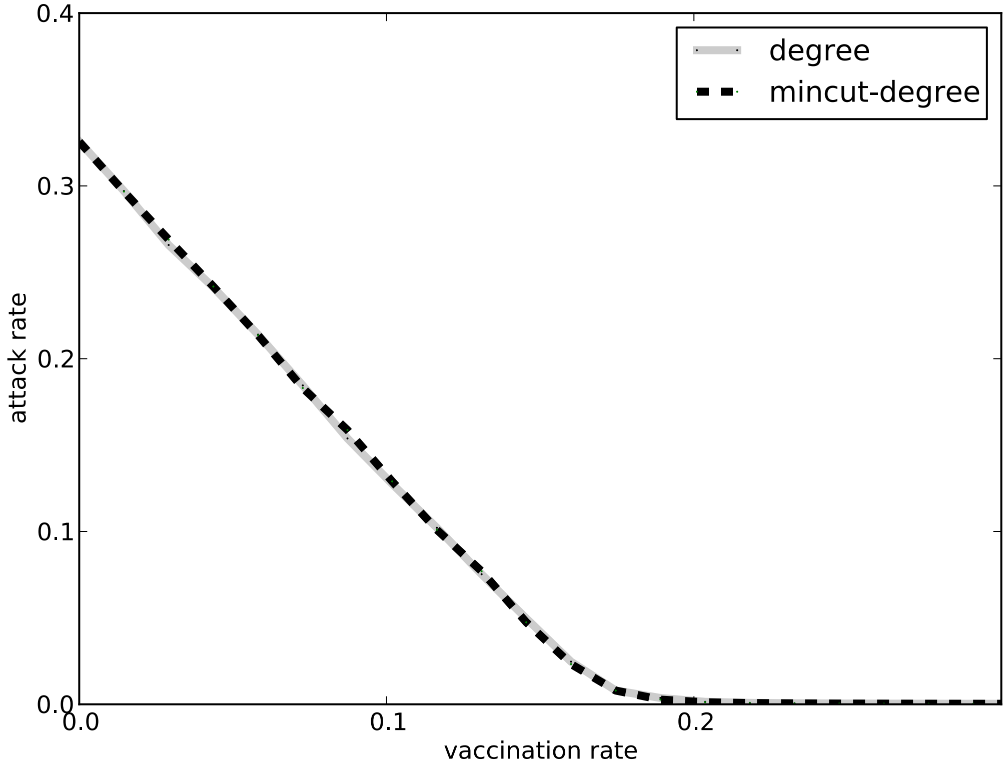


Figure 4.22: Comparison of the effectiveness of the mincut-degree-based policy and degree-based policy. Policies are generate from the $moderate_1$ graph for time window $T = 1$, peak transmission probability $p = .1$, and $thresh = .3$

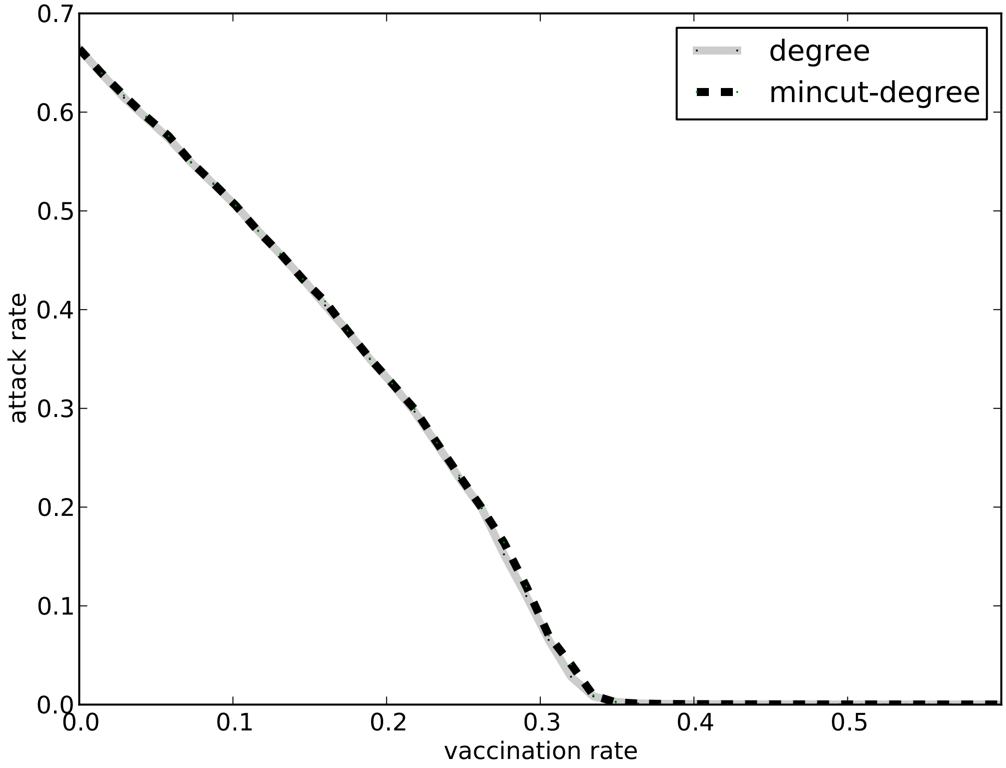


Figure 4.23: Comparison of the effectiveness of the mincut-degree-based policy, and degree-based policy. Policies are generate from the $moderate_1$ graph for time window $T = 1$, peak transmission probability $p = .5$, and $thresh = .5$

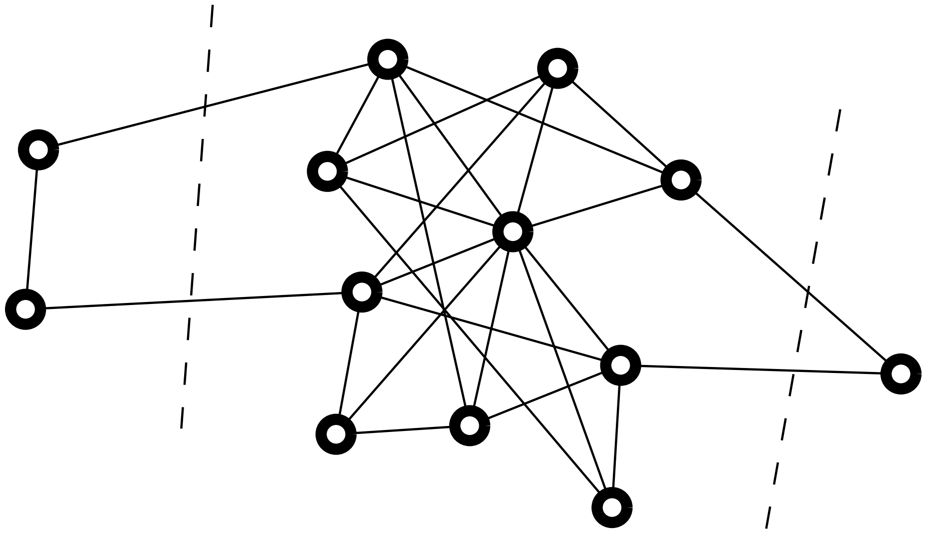


Figure 4.24: Trimming of boundary vertices as a result of the mincut percolation process.

exhibited by the HCW contact networks. Recall that positive assortativity, in general terms, means that high degree vertices to have edges with other high degree vertices. Newman [85] performed a number of experiments on random graph models with positive assortativity. His conclusion is that graphs with high assortativity exhibit a giant component at lower edge density. That is, if we think about the graph starting with no edges and growing, by adding more edges, graphs with high assortativity form a giant component at lower edge density than graphs with no, or negative, assortativity. Newman describes this as a "core group" of vertices which are very densely connected, with lower degree vertices that are simply "attached" to this core group. Further, his analysis suggest that assortative graphs are resilient to vertex removal and thus network connectivity of these assortative networks is not easily "destroyed". These claims would also explain the observations we see here and give further evidence that assortativity is an important property of our HCW contact networks. Also, positive assortativity means that a high degree vertex is unlikely to be connected to a lot of low degree vertices (i.e., those which would be "cut" by the mincut algorithm). Thus, these high degree vertices are unlikely to see a decrease in their degree based on this algorithm. As part of our investigation of the behavior we see here, we calculated the correlation between the degree of a vertex and the mean weight of adjacent edges and found a Pearson correlation coefficient of .05 with p-value 0.0002). This means that high degree vertices are more likely to have an incident edge of high weight.

CHAPTER 5
BUDGETED MAXIMUM COVERAGE

The Emerging Infections Network (EIN) (http://ein.idsociety.org/) is a "sentinel" network of clinical infectious disease specialists, primarily from the United States, created in 1995 by the Infectious Diseases Society of America with a Co-operative Agreement Program award from the Centers for Disease Control (CDC). The goal of the EIN is to assist the CDC and other public health authorities with surveillance of emerging infectious diseases and related phenomena (new treatment protocols, possible side effects of new vaccines, etc). To achieve its goal, the EIN maintains a private listserv open to infectious disease specialists, CDC investigators, and public health officials. There are currently over 1400 subscribers who receive roughly 3 emails per day. Since its inception, the EIN listserv has served over 2800 discussions on the identification of new infectious diseases, treatments, and policy implications.

There are a few features that distinguish the EIN listserv from other online mailing lists. Each submission (*post*) to the EIN listserv is sent to the EIN coordinator, a person responsible for managing the mailing list. The EIN coordinator is responsible for screening and filtering each post by fixing grammatical errors, providing links to citations, and removing any identifying patient information. Each post received by the EIN coordinator is either the start of a new *thread*, if that post is about a new topic, or a response to a previous post in an ongoing thread. Posts are collected throughout the day and bundled into a *mailing* which is broadcast to all subscribers the following morning.

Recent work at Google [45] (`http://www.google.org/flutrends/`) and Yahoo! Research [93] has focused on using search engine query terms as a means of tracking the spread of influenza. In the spring of 2009 as news of swine flu spread, numerous projects were initiated that used Twitter posts to track and observe the spread of the infection (see this project at Iowa [101] for an example). The EIN provides very different kind of information to public health officials compared to the large scale online efforts that attempt to tap into the "wisdom of the crowds." Even though the EIN is sometimes the first to detect or report an outbreak, its real utility comes later when clinical aspects of emerging infectious diseases get discussed. For example, in the spring of 2009 when news of the H1N1 virus was everywhere in the popular media, the EIN was relatively quiet on this topic. However, in late 2009 the EIN was buzzing with H1N1 related posts as doctors and public health officials get ready to deal with a large number of cases. EIN members were discussing not just the emergence or spread of H1N1, but its treatment, vaccine administration, patient care, etc. [1, 2, 3]. One EIN member posted their concern about H1N1 vaccine reacting to neural tissue and causing Guillain-Barré Syndrome (GBS), a rare disorder resulting in limb weakness and paralysis. One responder identified a possible case of this and another pointed to historical evidence supporting the original concern. Further discussion amplified these concerns and provided information to the CDC which instituted a case-finding protocol to monitor the situation, not only for GBS but for all immunization side-effects. Another EIN member identified a situation where healthcare workers were refusing to treat patients with H1N1 due to fear of exposure. Responders noted similar experiences, identified ethical concerns, and suggested policies. Occasionally discussion on the EIN can lead to discovery of previously unknown virus strains. For example, a post on the EIN in 2005 reported a number of severe pneumonia cases caused by the adenovirus, a common cause of respiratory illness [24]. Responses on the EIN mailing list helped identify these initial instances as a rare strain of community-acquired pneumonia which was previously unrecognized and later dubbed "the killer cold."

Identifying threads that are important is currently ad hoc, done by simply reading all the posts that make their way to the EIN. There is significant interest in improving the accuracy and timeliness with which this information is identified so that it can be distributed to the CDC and other healthcare organizations. Motivated by this need and the expectation that the EIN will grow in size in the near term, our goal is to develop a simple, low-cost procedure that can be used to *sample* traffic on the EIN and predict the emergence of important threads. Such a procedure will help focus the attention of doctors and public health officials to important, emerging discussions on the EIN. Ideally, we want to be able to identify threads that have the potential to become "important," and ignore threads that are "noise." Our approach is to look at historical EIN data (we have EIN data from Feb. 1997 to May 2009) and identify users who typically participate in the early stages of many important threads, but are involved in very few unimportant threads. If we are able to identify such "bellwether" users, then tracking these users can quickly point people who make policies to emerging important threads that are in their early stages of evolution, without inundating them with irrelevant information.

Suppose we have identified a set S of these "bellwether" users. Anyone wanting to identify important discussions, can follow this simple monitoring procedure:

> An unmarked thread t is marked "to be monitored" as soon as a member of S posts to t. Thread t is closely monitored until it dies or is deemed irrelevant.

The problem is then to find a set S of EIN participants who act as "bellwethers." That is, find a set S of users who participate in many important threads, but do not participate in many unimportant threads.

The above monitoring procedure presupposes a classification of threads into *important threads*, those that contain emerging phenomena worth closely following and *unimportant threads*, those that are irrelevant from the point of view of infectious disease concerns. This classification can be done in an automated manner or by consultation with a infectious disease expert. This classification can also be probabilistic: to each thread t we associate a probability $p(t)$ of being important (and therefore a probability $1 - p(t)$ of being unimportant). We also need a precise notion of *participation* in a thread. Since we are interested in early detection, we use a parameter m and say that a user u participates in a thread t if u makes a post to t within the first m mailings. Once these notions are defined precisely, we can associate with every subset S of users a *reward* $r(S)$ and a *cost* $c(S)$. $r(S)$ can be defined as the number of important threads in which users in S participate. In other words,

$r(S)$ is the number of important threads that will be monitored if the set S of users is tracked. $c(S)$ can be defined as the number of unimportant threads in which users in S participate. In other words, $c(S)$ is the number of unimportant threads that will have to be monitored if the set S of users is tracked. More general definitions of reward and cost are possible; for example, we could associate with each thread t a weight $w(t)$ and define $r(S)$ as the sum of the weights of important threads in which users in S participate. The definition of $c(S)$ can be generalized in a similar manner. If the notion of important and unimportant threads is defined probabilistically, then the definitions of reward and cost can be extended to refer to expected values. In this setting, good choices for S are obtained by solving the following budgeted maximization problem:

$$\max_{S \subseteq U} \; r(S) \qquad \text{s. t.} \qquad c(S) \leq B$$

Here U is the set of all users and B is a given cost budget.

All of the different versions of the reward function $r : 2^U \rightarrow \mathbb{R}^+$ mentioned above are *submodular*. Recall that a function $f : 2^U \rightarrow \mathbb{R}^+$ is said to be *submodular* if $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ forall $A, B \subseteq U$. The problem of maximizing submodular set functions has a long history dating back to the 70's [78]. In their seminal work, Nemhauser et al. [78] consider the problem of maximizing a submodular set function $f : 2^U \rightarrow \mathbb{R}^+$ subject to a cardinality constraint. They show that a simple greedy algorithm yields a $(1 - \frac{1}{e})$-approximation. Calinescu et al. extend this result to the problem of maximizing a monotone submodular set function subject to a matroid constraint and gave a randomized $(1 - \frac{1}{e})$-approximation algorithm [21].

For a knapsack constraint Khuller et al. [55] show the simple greedy algorithm also yields a $(1 - \frac{1}{e})$-approximation. More recently these results have been extended to problems with multiple constraints [63, 59].

Most relevant to our work is the work of Khuller et al. [55] who suppose that each element $u \in U$ is associated with a *cost* $c(u)$ and the *cost* $c(S) = \sum_{u \in S} c(u)$. Their problem is to find a subset $S \subseteq U$ with maximum $f(S)$ from among all sets $S \subseteq U$ satisfying $c(S) \leq B$, they call this the Budgeted Maximum Coverage (BMC) problem. The BMC problem has been used in applications similar to ours by Leskovec et al. [65] and El-Arini et al. [34] to monitor the blogosphere.

Our budgeted maximization problem turns out to be fundamentally different on account of its cost structure. Because a thread is only flagged once, the cost $c(\{u, u'\})$ of monitoring two users $u, u' \in U$, could be much smaller than $c(u) + c(u')$ because of substantial overlap in the unimportant threads in which u and u' participate. Later we consider the greedy algorithm of Khuller et al. [55] that yields a constant-factor approximation for the BMC problem and construct a simple instance of our problem for which this greedy algorithm performs arbitrarily poor. We also show a reduction from the *densest k-subhypergraph* problem [49] to our problem which indicates that our problem cannot be approximated within a factor of $O(2^{(\log(n))^{\delta}})$ for some $\delta > 0$ under the assumption that $3 - SAT \notin DTIME(2^{n^{\frac{3}{4}+\epsilon}})$.

5.0.2 Results

We model the problem of monitoring a listserv, such as the EIN, as a type of budgeted maximum coverage problem. Even though our problem seems superficially similar to the budgeted maximum coverage problem considered by Khuller et al. [55], from an algorithmic point of view they are fundamentally different. The budget constraint of Khuller et al. [55] is linear, whereas ours is not. We show that the simple greedy algorithm that works well for the problem of Khuller at al. performs arbitrarily poor on some instances of our problem. Furthermore, by showing a reduction from the *densest k-subhypergraph* [49] problem we show that in general our problem cannot be approximated within a factor of $O(2^{(\log(n))^{\delta}})$ for some $\delta > 0$ under the assumption that $3-SAT \notin DTIME(2^{n^{\frac{3}{4}+\epsilon}})$. Nevertheless, experimental runs of the greedy algorithm on the EIN data show that greedy performs remarkably well relative to OPT. We identify a possible feature of our EIN data, that we call the *overlap condition*, and show that the greedy algorithm does indeed provide a constant-factor approximation guarantee if the overlap condition is satisfied. Using an implementation of our greedy algorithm on the EIN data, we select a set of "bellwether" users to track and reduce the work involved in monitoring the EIN for a year by over 75%. Additionally, we validate our experiments by showing that the set of users we select will flag all of the important threads and the number of "noisy" threads (false-positives) is low.

5.1 The Reward-Cost Model

Let T denote the set of threads, U denote the set of users, and $G = (T, U, E)$ denote the *user-thread graph*, a bipartite graph with edges $\{u, t\}$, $u \in U$, $t \in T$, whenever user u *participates* in thread t. We will make the notion of participation precise later. For any $u \in U$, let $N(u)$ denote the threads that user u participates in and for any subset $S \subseteq U$ of users let $N(S) = \cup_{u \in S} N(u)$. Associated with each thread $t \in T$, there is a probability $p(t)$ of thread t being important and a positive weight $w(t)$. For any subset $S \subseteq U$ of users, we define the set functions $r : 2^U \to \mathbb{R}^+$ and $c : 2^U \to \mathbb{R}^+$ as:

$$\begin{aligned} r(S) &= \sum_{t \in N(S)} p(t) \cdot w(t) \\ c(S) &= \sum_{t \in N(S)} (1 - p(t)) \cdot w(t) \end{aligned}$$

In this paper we focus on the deterministic setting where $p(t) \in \{0, 1\}$ for each $t \in T$ and use T^+ to denote *important* threads, i.e., those threads t with $p(t) = 1$, and T^- to denote *unimportant* threads, i.e., those threads t with $p(t) = 0$. For ease of exposition we assume $w(t) = 1$ for all $t \in T$. The *budgeted maximization problem with overlapping costs* (BMOC) problem takes as input a user-thread graph $G = (U, T, E)$, probabilities $p : T \to [0, 1]$, weights $w : T \to \mathbb{R}^+$, a $B \in \mathbb{R}^+$ and aims to find a subset $S \subseteq U$ that maximizes $r(S)$ while satisfying the budget constraint $c(S) \leq B$.

5.1.1 Choosing Important Threads

One way to classify threads into important and unimportant threads is to consult infectious disease specialists. For example, one might survey EIN subscribers or

Threads	Users	Posts	Mailings per thread			Posts per thread			People per thread		
			Avg.	Min.	Max.	Avg.	Min.	Max.	Avg.	Min.	Max.
2833	1451	13,502	2.85 ±1.91	1.00	18.00	4.77 ±4.62	1.00	58.00	4.417 ±3.98	1.0	34.00

Figure 5.1: Summary statistics for all EIN traffic from February 1997 through May 2009 including the average (± standard deviation), minimum, and maximum number of mailings, number of posts, and number of users per thread.

have an online rating system in place. Since these approaches suffer from low response rate and are not currently in place, we develop an automated procedure for picking important threads by assuming that any thread worth monitoring closely will have sufficient EIN activity and therefore such threads can be identified by characteristics such as (a) number of mailings, (b) number of posts, and (c) number of distinct participants. Summary statistics of threads with respect to each of these characteristics are shown in Figure 5.1. The distributions of the number of threads with respect to each of these characteristics are heavy-tailed.

One simple way to pick "important" threads by paying attention to all three characteristics is the following. Let M^* be the maximum number of mailings in any thread, P^* be the maximum number of posts in any thread, and D^* be the maximum number of distinct participants in any thread (see Figure 5.1). Further, let $M(t)$, $P(t)$, and $D(t)$ denote the number of mailings, posts, and distinct users for thread t respectively. For a threshold $thresh$, $0 \leq thresh \leq 100$, we let $T^+(thresh) = \{t \in T | M(t) \geq thresh \cdot M^*\} \cap \{t \in T | P(t) \geq thresh \cdot P^*\} \cap \{t \in T | D(t) \geq thresh \cdot D^*\}$. Simply put, the important threads are threads that have higher than $thresh$ percentage of the maximum value for mailings, posts, and distinct users.

Figure 5.2(a) shows the cardinality of $T^+(thresh)$ for each $thresh$, $0 \leq thresh \leq 100$.

5.1.2 Criteria for Participation

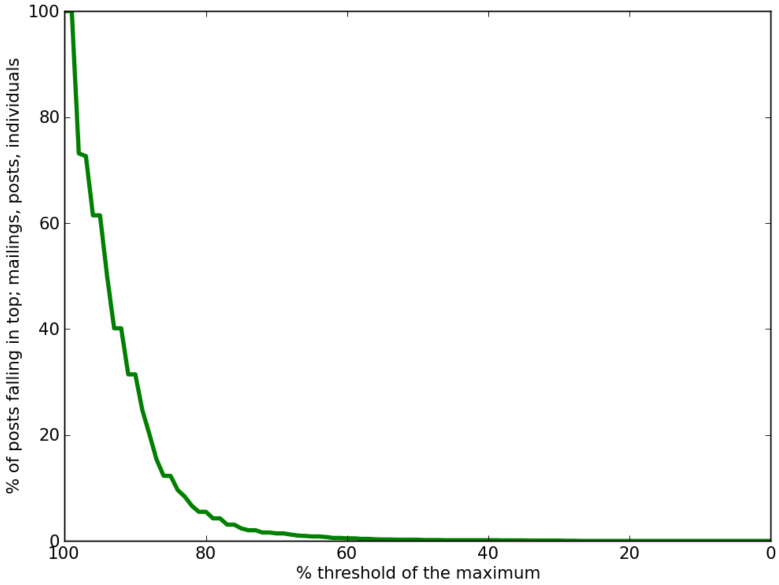
Since we are interested in early detection of potentially interesting threads, we focus on posts to a thread that are made very early on in life the thread. Specifically, given a parameter m we say that a user u *participates* in a thread t if u posts to t with the first m mailings of t. In our experiments, we use values $1, 2$, and 3 for m.

5.2 A Greedy Algorithm for BMOC

Khuller et al. [55] present a simple greedy algorithm for the budgeted maximum coverage problem in which the budget constraint is linear and show that this algorithm guarantees a $\frac{1}{2}\left(1 - \frac{1}{e}\right)$-factor approximation ratio. When combined with an enumeration technique, this algorithm provides a $\left(1 - \frac{1}{e}\right)$-factor approximation ratio. To state this greedy algorithm in the context of our problem, we need notation for incremental reward and cost of adding a user to our current solution. Let $S \subseteq U$ and $u \in U \setminus S$. Then,

$$\begin{aligned} r(S,u) &= |\{t \in T^+ \mid t \notin N(S), t \in N(u)\}| \\ c(S,u) &= |\{t \in T^- \mid t \notin N(S), t \in N(u)\}| \end{aligned}$$

Algorithm 5.1 gives pseudocode for the greedy algorithm, which we call **Greedy**, combining two algorithms, which we call **GreedyRatio** and **GreedyReward**. **GreedyRatio** starts with an empty set S of users and repeatedly adds to S a user u who



(a)

thresh		Imp.	Unimp.
60%	Mean Mailings	12.00	2.79
	Mean # Distinct Users	25.11	4.28
	Mean # Posts	33.28	4.58
70%	Mean Mailings	9.88	2.73
	Mean # Distinct Users	20.23	4.14
	Mean # Posts	25.06	4.41
80%	Mean Mailings	6.98	2.56
	Mean # Distinct Users	14.79	3.69
	Mean # Posts	16.95	3.91

(b)

Figure 5.2: Data on classification of important threads. (a) percentage of threads whose number of mailings, number of posts, and number of participants are all within *thresh* % of the corresponding maximum values of these characteristics. Since $T^+(thresh) \supseteq T^+(thresh')$ for $thresh > thresh'$, we obtain larger sets of important threads as we increase *thresh* from 60 to 80. With $x = 60$, we pick up 18 (out of 2818) important threads, with $x = 70$, we pick up 47 (out of 2818) important threads, and with $x = 80$, we pick up 183 (out of 2818) important threads. (b) Aggregate statistics of important threads for different values of *thresh*.

maximizes $\frac{r(S,u)}{c(S,u)}$ and whose addition to S does not violate the budget constraint. Similarly, **GreedyReward** starts with an empty set S of users and repeatedly adds to S a user u who maximizes $r(S,u)$ and whose addition to S does not violate the budget constraint. Let S' be the output of **GreedyRatio** and S'' be the output of **GreedyReward**. The algorithm **Greedy** runs **GreedyRatio** and **GreedyReward** and returns either S' or S'', whichever has the greater reward.

It is easy to construct an instance of BMOC for which **Greedy** performs arbitrarily poorly (see Figure 5.3).

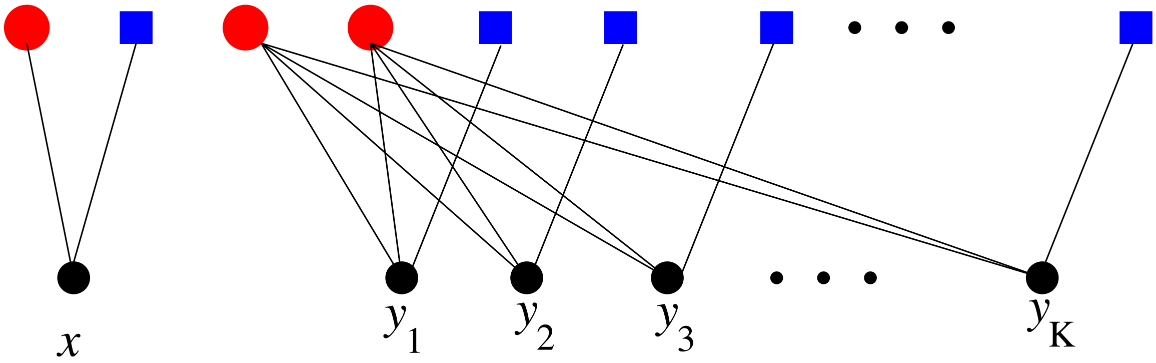


Figure 5.3: A user-thread graph with red vertices (circles) denoting unimportant threads and blue vertices (squares) denoting important threads. For this instance with budget $B = 2$, **Greedy** will pick x and obtain a reward of 1, whereas the optimal solution consists of $\{y_1, y_2, \ldots, y_K\}$ for a reward of K.

5.2.1 BMOC is Difficult to Approximate

Bad news about BMOC is that a special case of BMOC is at least as hard as the **Densest** k**-SubHypergraph** problem. The **Densest** k**-SubHypergraph** (DKSH) problem [49] takes as input a hypergraph $G = (V, E)$ and seeks to find

Algorithm 5.1 Greedy Algorithm for BMOC

```
GreedyRatio($U$)
$S' \leftarrow \emptyset$
$U' \leftarrow U$

while $U' \neq \emptyset$
    Pick $u \in U'$ that maximizes: $\frac{r(S',u)}{c(S',u)}$
    if $c(S' \cup \{u\}) \leq B$
        $S' \leftarrow S' \cup \{u\}$
    end if
    $U' \leftarrow U' \setminus \{u\}$
end while
return $S'$

GreedyReward($U$)
$S'' \leftarrow \emptyset$
$U' \leftarrow U$

while $U' \neq \emptyset$
    Pick $u \in U'$ that maximizes: $r(S'', u)$
    if $c(S'' \cup \{u\}) \leq B$
        $S'' \leftarrow S'' \cup \{u\}$
    end if
    $U' \leftarrow U' \setminus \{u\}$
end while
return $S''$

Greedy($U$)
$S' \leftarrow$ GreedyRatio($U$)
$S'' \leftarrow$ GreedyReward($U$)
if $r(S') \geq r(S'')$
    return $S'$
else
    return $S''$
end if
```

a subset of k vertices that induce a subhypergraph of G with maximum number of edges. Assuming that $3-SAT \notin DTIME(2^{n^{\frac{3}{4}+\epsilon}})$, DKSH is inapproximable to within a factor of $O(2^{(\log(n))^{\delta}})$, $\delta > 0$. Further, DKSH is a generalization of the k-**Densest Subgraph** (KDS) problem [40] for which the currently best known algorithm yields an $O(n^{\alpha})$-approximation where $\alpha < \frac{1}{4}$ [15]. Improving this approximation factor is an important open problem in the area of approximation algorithms. There is a simple reduction from DKSH to BMOC, originally sketched by Chekuri [25], which shows that a β-approximation algorithm for BMOC will imply a β-approximation for DHSH and subsequently KDS. Using this reduction we prove the following theorem.

Theorem 5.1. *If there is a β-approximation algorithm for BMOC, there is a β-approximation algorithm for* ***Densest*** k***-SubHypergraph****.*

Proof. Start with an instance $\{G = (V, E), k\}$ of **Densest** k**-SubHypergraph** and construct a user-thread graph H with thread set $T = V \cup E$ and user set $U = E$. Designate E to be the important threads T^{+} and V to be the unimportant threads T^{-}. Corresponding to each hyperedge $e = \{u_1, \ldots, u_i\}$, connect each user $e \in U$ to unimportant threads $u_1, \ldots, u_i$ and important thread e. Set the budget B to k. Let us call a solution $S \subseteq U$ *maximal* if for all users $u \in U \setminus S$, $c(S \cup \{u\}) > c(S)$. By this definition we get the following claim,

Claim 1. *G has an induced subhypergraph with k vertices and m edges iff H has a maximal subset S of users with $c(S) = k$ and $r(S) = m$.*

Proof. Fix a set of k vertices in G and suppose they induce a subhypergraph G' of m edges. Let S denote the set of m edges in G' thus $c(S) = k$ and $r(s) = m$ by the construction of H. Now suppose that S is not maximal. Then there exists at least one user $e \in U$, corresponding to an edge in G which can be added to S such that $c(S \cup \{e\}) = c(S) = k$ and $r(S \cup \{e\}) = m + 1$. This means G' has $m+1$ edges which is a contradiction and thus S is maximal.

Now let S be a maximal subset of users in H with $c(S) = k$ and $r(S) = m$. By the definition of the cost function c there are k unimportant threads connected to S corresponding to k nodes in G. Let G' be the subhypergraph induced by these k nodes. Because S is maximal, there is no user e, corresponding to an edge in G, which can be added to S without $c(S \cup \{e\}) > c(S)$. By construction of H, each user e corresponds to a single important thread e and since $r(S) = m$, the induced subgraph G' has exactly m edges.

Now suppose there exists a β-approximation algorithm A for BMOC. Start with an instance $\{G, k\}$ of the DKSH, transform it as specified above to an instance of BMOC $\{H, k\}$ and run A on it. The solution generated is a set of users S such that

$$c(S) \leq k, \qquad r(S) \geq \beta \cdot OPT$$

where OPT is the maximum reward of a subset S^* of users in H satisfying $c(S^*) \leq B$. Note that $|S^*| = r(S^*) = OPT$, since each user is connected to exactly one important thread in H by the reduction from G. Without loss of generality suppose that both S and S^* are maximal. By the above claim, S^* corresponds to a size OPT set of edges in G induced by a set of no more than k vertices V'^*. Similarly, S corresponds to a set of edges in G induced by a set of no more than k vertices, V'. Note that S^* is an optimal solution to BMOC iff V'^* is an optimal solution to DKSH. Thus OPT is maximum number of edges induced by any subset of no more than k vertices in G. And since $|S| \geq \beta \cdot |S^*| = \beta \cdot OPT$, V' induces a subgraph of size no less than $\beta \cdot OPT$.

5.2.2 The Overlap Condition

In the bad example for the greedy algorithm in Figure 5.3, the unimportant threads have high average degree (i.e., $(2K+1)/3$) relative to the average degree of important threads (which is just 1). While this is possible in general for BMOC, our specific criteria for identifying important and unimportant threads from the EIN data makes this unlikely for our instances of the problem. We now formalize this heuristic notion, calling it the *overlap* condition and show that if we assume that the overlap condition holds, then **Greedy** provides a $\frac{1}{2}(1-\frac{1}{e})$-approximation. In fact, assuming the overlap condition we can obtain a $(1-\frac{1}{e})$-approximation by using **Greedy** in combination with the enumeration technique described by Khuller et al. [55].

Let S_i denote the set of the first i users selected by **GreedyRatio**. Denote the remaining users, i.e., $U \setminus S_i$ as U_i and let G_i denote the bipartite graph obtained from the user-thread graph G by deleting $S_i \cup N(S_i)$. Thus the users in G_i are those in U_i and the threads in G_i are those that are not "covered" by users in S_i. For any subset $U' \subseteq U_i$ of users, let $G_i[U']$ denote the bipartite subgraph of G_i induced by $U' \cup N(U')$. Let $\delta^+(i, U')$ (respectively, $\delta^-(i, U')$) denote the average degree of the important (respectively, unimportant) threads in $G_i[U']$. We define the *overlap* condition as:

$$\forall i, \forall U' \subseteq U_i : \delta^+(i, U') \geq \alpha_i \cdot \delta^-(i, U') \tag{5.1}$$

where α_i is a constant. Let $r(S_i, U')$ (respectively, $c(S_i, U')$) denote the number of important (respectively, unimportant) threads in $N(U') \setminus N(S_i)$. It is easy to verify that

$$\begin{aligned} \delta^+(i, U') &= \frac{\sum_{u \in U'} r(S_i, u)}{r(S_i, U')} \\ \delta^-(i, U') &= \frac{\sum_{u \in U'} c(S_i, u)}{c(S_i, U')}. \end{aligned}$$

and therefore the overlap condition can be equivalently stated as

$$\forall i, \forall U' \subseteq U_i : \frac{\sum_{u \in U'} r(S_i, u)}{r(S_i, U')} \geq \alpha_i \cdot \frac{\sum_{u \in U'} c(S_i, u)}{c(S_i, U')}. \tag{5.2}$$

By definition of the overlap condition, after i users have been chosen,

$$\alpha_i = \min_{U' \subseteq U_i} \frac{\delta^+(i, U')}{\delta^-(i, U')}.$$

Note that α_i is bounded above by 1 because when $U' = \{u\}$, a single user, $\delta^+(i, U') = \delta^-(i, U') = 1$.

Let OPT be an optimal set of users. Suppose that after some number of iterations, **GreedyRatio** has selected a set S of users. In the next iteration, **GreedyRatio** considers an element $u \notin S$ that maximizes $\frac{r(S,u)}{c(S,u)}$. This element may or may not be added to S depending on whether adding u to S causes the budget constraint to be violated. Suppose that r is the number of iterations executed by **GreedyRatio** until the first user $u \in OPT$ is considered, but rejected (due to violation of the budget constraint). Suppose that ℓ users have been selected by **GreedyRatio** during these r iterations. Label these users $u_1, u_2, \ldots, u_\ell$ in the order in which they were selected by **GreedyRatio** and let $u_{\ell+1}$ be the first user in OPT considered but rejected. Let j_i be the iteration in which user u_i was considered and let $S_0 = \emptyset$, $S_i = S_{i-1} \cup \{u_i\}$ for each $i = 1, 2, \ldots, \ell$.

The following lemma uses the overlap condition to extend the key lemma in [55] to instances of BMOC in which the overlap constraint holds. The calculations in the subsequent lemmas are similar to those in [55] but are included mainly for completeness.

Lemma 5.2. *If the overlap condition is satisfied, then after each iteration $j_i, i = 1, 2, \ldots, \ell + 1$,*

$$r(S_{i-1}, u_i) \geq \alpha_{i-1} \cdot \frac{c(S_{i-1}, u_i)}{B} \Big(r(OPT) - r(S_{i-1}) \Big).$$

Proof. For each user $u \in OPT \setminus S_{i-1}$, due to the greedy choice of u_i, the ratio $\frac{r(S_{i-1},u)}{c(S_{i-1},u)}$ is at most $\frac{r(S_{i-1},u_i)}{c(S_{i-1},u_i)}$. Therefore,

$$\frac{\sum_{u \in OPT \setminus S_{i-1}} r(S_{i-1}, u)}{\sum_{u \in OPT \setminus S_{i-1}} c(S_{i-1}, u)} \leq \frac{r(S_{i-1}, u_i)}{c(S_{i-1}, u_i)}. \tag{5.3}$$

According to the overlap condition,

$$\frac{\sum_{u \in OPT \setminus S_{i-1}} r(S_{i-1}, u)}{\sum_{u \in OPT \setminus S_{i-1}} c(S_{i-1}, u)} \geq \alpha_{i-1} \cdot \frac{r(S_{i-1}, OPT \setminus S_{i-1})}{c(S_{i-1}, OPT \setminus S_{i-1})}.$$

Combining this with (5.3) yields

$$\alpha_{i-1} \cdot \frac{r(S_{i-1}, OPT \setminus S_{i-1})}{c(S_{i-1}, OPT \setminus S_{i-1})} \leq \frac{r(S_{i-1}, u_i)}{c(S_{i-1}, u_i)}. \tag{5.4}$$

Substituting into the above inequality the fact that $c(S_{i-1}, OPT \setminus S_{i-1}) \leq c(OPT) \leq B$, we get

$$r(S_{i-1}, OPT \setminus S_{i-1}) \leq \frac{B}{\alpha_{i-1}} \cdot \frac{r(S_{i-1}, u_i)}{c(S_{i-1}, u_i)}$$

Further, $r(OPT) - r(S_{i-1})$ is at most $r(S_{i-1}, OPT \setminus S_{i-1})$ which leads to

$$r(OPT) - r(S_{i-1}) \leq \frac{B}{\alpha_{i-1}} \cdot \frac{r(S_{i-1}, u_i)}{c(S_{i-1}, u_i)}$$

Moving terms around, yields the lemma.

Lemma 5.3. *If the overlap condition holds, then for iterations $j_i, i = 1, 2, \ldots, \ell + 1$,*

$$r(S_i) \geq \left[1 - \prod_{k=1}^{i} \left(1 - \alpha_{i-1} \frac{c(S_{k-1}, u_k)}{B}\right)\right] r(OPT)$$

Proof. The proof follows by induction on the iterations $j_i, i = 1, 2, \ldots, \ell + 1$. The base case j_1 comes from setting $i = 1$ and using Lemma 5.2 which gives us

$$r(S_1) = r(S_0, u_1) \geq \alpha_0 \frac{c(S_0, u_1)}{B} r(OPT).$$

Assuming the lemma holds for iterations $j_i, i = 1, .., i - 1$ we show it holds for j_i:

$$\begin{aligned}
r(S_i) &= r(S_{i-1}) + r(S_{i-1}, u_i) \\
&\geq r(S_{i-1}) + \alpha_{i-1} \frac{c(S_{i-1}, u_i)}{B} \left(r(OPT) - r(S_{i-1})\right) \\
&= \left[1 - \alpha_{i-1} \frac{c(S_{i-1}, u_i)}{B}\right] r(S_{i-1}) + \alpha_{i-1} \frac{c(S_{i-1}, u_i)}{B} r(OPT) \\
&\geq \left[1 - \alpha_{i-1} \frac{c(S_{i-1}, u_i)}{B}\right] \cdot \left[1 - \prod_{k=1}^{i-1} \left(1 - \alpha_{i-1} \frac{c(S_{k-1}, u_k)}{B}\right)\right] r(OPT) + \\
&\quad \alpha_{i-1} \frac{c(S_{i-1}, u_i)}{B} r(OPT) \\
&= \left[1 - \prod_{k=1}^{i} \left(1 - \alpha_{i-1} \frac{c(S_{k-1}, u_k)}{B}\right)\right] r(OPT)
\end{aligned}$$

where the first inequality follows from Lemma 5.2 and the second inequality follows from inductive hypothesis.

Theorem 5.4. *If an instance of the user-thread graph $G = (U, T, E)$ satisfies the overlap condition with respect to ℓ iterations of the Algorithm* ***GreedyRatio*** *then the set S of users returned by Algorithm* ***Greedy*** *satisfies*

$$r(S) \geq \frac{1}{2}\left(1 - \frac{1}{e^{\alpha}}\right) \cdot OPT,$$

where OPT is the maximum reward associated with any set of users whose cost is at most the budget B and α is the average value of α_i, $i = 0, \ldots, \ell$.

Proof. Consider iteration $\ell + 1$. Using lemma 5.3 and the fact that $c(S_{\ell+1}) > B$ we have:

$$\begin{aligned} r(S_{\ell+1}) &\geq \left[1 - \prod_{k=1}^{\ell+1}\left(1 - \alpha_{k-1}\frac{c(S_{k-1}, u_k)}{B}\right)\right] r(OPT) \\ &\geq \left[1 - \prod_{k=1}^{\ell+1}\left(1 - \alpha_{k-1}\frac{c(S_{k-1}, u_k)}{c(S_{\ell+1})}\right)\right] r(OPT) \\ &\geq \left[1 - \left(1 - \frac{\alpha}{\ell+1}\right)^{\ell+1}\right] r(OPT) \\ &\geq \left(1 - \frac{1}{e^{\alpha}}\right) r(OPT). \end{aligned}$$

The third inequality follows from the fact that

$$\left[1 - \prod_{k=1}^{\ell+1}\left(1 - \alpha_{k-1}\frac{c(S_{k-1}, u_k)}{c(S_{\ell+1})}\right)\right]$$

has minimum value $1 - (1 - \alpha/(\ell+1))^{\ell+1}$ when $\alpha_0 c(S_0, u_k) = \cdots = \alpha_\ell c(S_\ell) = \alpha c(S_{\ell+1})/(\ell+1)$ and $\alpha = \frac{\sum_{k=1}^{\ell+1}\alpha_{k-1}}{\ell+1}$. Thus,

$$r(S_{\ell+1}) = r(S_\ell) + r(S_\ell, u_{\ell+1}) \geq (1 - \frac{1}{e^{\alpha}})r(OPT)$$

Note that the reward given by **GreedyReward**, $r(S') \geq r(S_\ell)$, and since $r(S_0, u_{\ell+1})$ is at most the maximum reward for a single user, the reward given by **GreedyReward**, $r(S'') \geq r(S_0, u_{\ell+1})$. This gives us:

$$r(S') + r(S'') \geq r(S_\ell) + r(S_\ell, u_{\ell+1}) \geq \left(1 - \frac{1}{e^\alpha}\right) r(OPT)$$

Therefore either the reward given by **GreedyRatio**, $r(S')$ or the reward given by **GreedyReward**, $r(S'')$ is greater than or equal to $\frac{1}{2}\left(1 - \frac{1}{e^\alpha}\right) r(OPT)$.

Notice that the α term denotes the average α_i over every iteration i of the **GreedyRatio** algorithm. Theorem 5.4 still holds as long as the overlap condition is satisfied *on average* by a constant factor α.

We "tested" the overlap condition for the EIN data in a limited way by considering all pairs and triples of users (see Table 5.1). Specifically, when $i = 0$, $S_i = \emptyset$, $U_i = U$ the overlap condition reduces to

$$\forall U' \subseteq U : \delta^+(U') \geq \alpha_0 \cdot \delta^-(U'),$$

where $\delta^+(U')$ (respectively, $\delta^-(U')$) is the average degree of the important threads (respectively, unimportant threads) in the subgraph of G induced by $U' \cup N(U')$.

5.3 Experiments on BMOC

Choosing a particular threshold *thresh* (60, 70, or 80), as described in Section 5.1.1, induces a partition of the set of threads into important and unimportant threads. By fixing a value for the participation parameter m (1, 2, 3, or ∞), as described in Section 5.1.2, we fix the threads in which each individual participated.

Table 5.1: Results from analyzing the overlap condition for user subsets $U' \subseteq U$ of *pairs* ($|U'| = 2$) and *triples* ($|U'| = 3$) in the user-thread graph with *thresh* $= 80$.

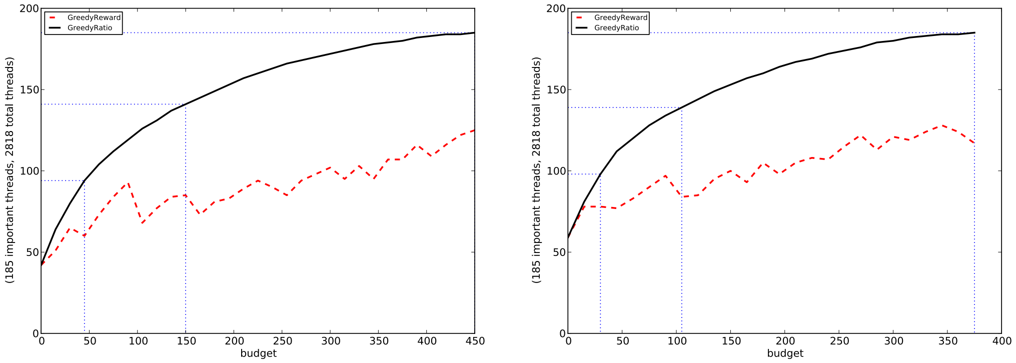
	Pairs	Triples
Total	271784	68456236
OC Holds	271490(99.98%)	68443062(99.98%)
Min. Factor (α_0)	0.704	0.647
Avg. Factor (α_0)	2.96	2.90

OC Holds shows the number of sets for which $\delta^+(U') \geq \delta^-(U')$. *Min. Factor* and *Avg. Factor* show the smallest value and average value of $\frac{\delta^+(U')}{\delta^-(U')}$ over all U'.

Having fixed *thresh* and m, we consider all values of the budget B, starting with $B = 0$, until we achieve full coverage of all important threads. Fixing values for *thresh*, m, and B creates an instance of BMOC that we use as input to **Greedy**.

5.3.1 Greedy Performance

Figure 5.4 shows plots for solutions found by **GreedyRatio** and **GreedyReward** for instances with *thresh* $= 80$ and participation parameter values $m = 2$ and $m = 3$. Recall that **Greedy** simply returns the better of the solutions produced by **GreedyRatio** and **GreedyReward**. Results shown here are similar for all *thresh* and m values we considered. We can view the reward of a solution returned by **GreedyRatio** or **GreedyReward** as a function of B. Note that neither of these functions are monotonic in B – simple examples are easy to construct for both algorithms. As a result, one simple improvement to these algorithms is to consider all values $B' = 1, 2, \ldots, B$ as the budget, run **GreedyRatio** and **GreedyReward** with each value of B' as the budget, and return as a solution, the subset that has maximum reward over all values of B'. Table 5.2 focuses on specific points on the plots in Figure 5.4, analyzing these more closely. In particular, this analysis focuses on points that provide 50%, 75%, and 100% coverage of the important threads.



(a) (b)

Figure 5.4: Plots showing the reward of solutions produced by **GreedyRatio** and **GreedyReward** with *thresh* = 80 and (a) $m = 2$ and (b) $m = 3$. The x-axis shows the budget and the y-axis shows reward. The black line shows the reward from **GreedyRatio** and the dashed line shows the reward from **GreedyReward**. The dotted lines mark points of interest, corresponding to 50%, 75%, and 100% coverage of important threads, discussed further in Table 5.2.

5.3.2 Analysis of Selected Users

The majority of active users on the EIN are doctors either in private practice, with only clinical responsibilities, or at an academic institution, where they have clinical and research responsibilities. Table 5.3(a) shows the distribution of users selected by **Greedy** (for *thresh* = 80 and full coverage) by whether they are at an academic institution, in private practice, or elsewhere. These results nicely match the expectations of a third party infectious disease expert [94]; doctors in private practice

Table 5.2: The cost of solutions that achieve 50%, 75%, and 100% coverage of important threads (corresponding to points from the plots shown in Figure 5.4).

thresh	m	c		
		50%	75%	100%
80% (185)	1	164.0	404.0	949.0
	2	45.0	148.0	436.0
	3	30.0	103.0	363.0
	∞	15.0	60.0	205.0

The key findings reported in this table are (a) the cost of full (respectively, 75%) coverage is roughly 10 (respectively, 3) times the cost of half coverage and (b) relaxing the requirement of early detection (i.e., increasing m from 1 to 3) decreases costs significantly.

tend to initiate more important threads, possibly because they have more clinical experience and have fewer colleagues with whom they can discuss issues face-to-face. Such users tend to turn to the EIN more frequently with important concerns. On the other hand first responders and later responders in important threads tend to be evenly distributed between doctors at academic institutions and those in private practice. Table 5.3(b) shows that selected users (at *thresh* = 80, full coverage) are geographically spread out quite evenly across the U.S. even though geographic coverage was not a criteria used in our algorithms. Figure 5.5 shows the selected EIN participants overlaid on a map of the United States.

There is also anecdotal evidence that users selected by our algorithm are indicators of valuable information on the EIN. We gave the EIN manager a list of names generated by **Greedy**, for threshold of $T = 80$ and $m = 3$, and one of the names on the list raised a red flag for being unrecognizable. After reviewing some of their older posts, the user was discovered to be an "untapped asset" to the EIN and is now being

Table 5.3: Statistics for sample runs of **Greedy** algorithm.

(a)

m	Total	Academic	Private	Other	Unknown
1	126	30(32.97%)	57(62.64%)	4(4.40%)	35
2	161	34(45.33%)	34(45.33%)	7(9.93%)	86
3	158	36(48.65%)	32(43.24%)	6(8.11%)	84
∞	186	33(42.86%)	35(45.45%)	9(11.69%)	109

(b)

m	Avg Distance ($\pm$)	Max Distance
2	230.57($\pm$169.23)	885.08
3	212.89($\pm$141.28)	714.59

(a) Distribution of users selected by **Greedy** (with *thresh* = 80, full coverage) by whether they are at an academic institution, private practice, or elsewhere. The column Total shows the total number of users selected by our algorithm.
(b) The geographic spread of users selected by **Greedy** (with *thresh* = 80, full coverage) is shown here. For example, with $m = 2$, every point in the continental U.S. is within 231 miles of a selected user, on average. These statistics were obtained by sampling 10 million points uniformly at random; more accurate results can be obtained by constructing Voronoi diagrams.

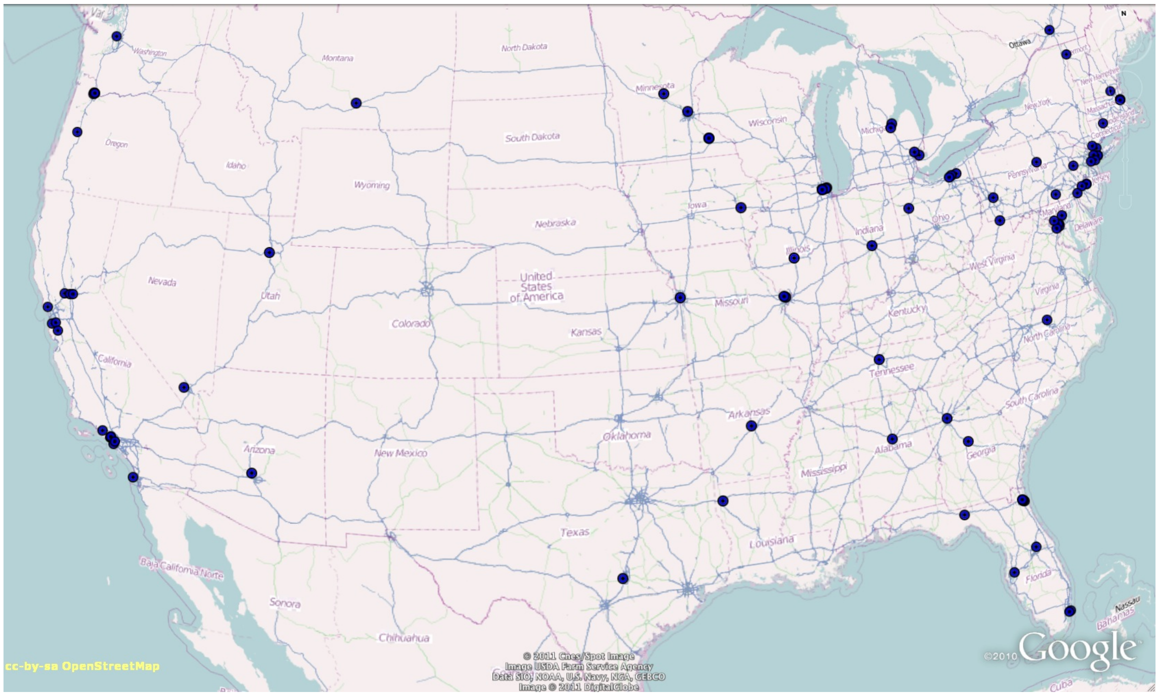
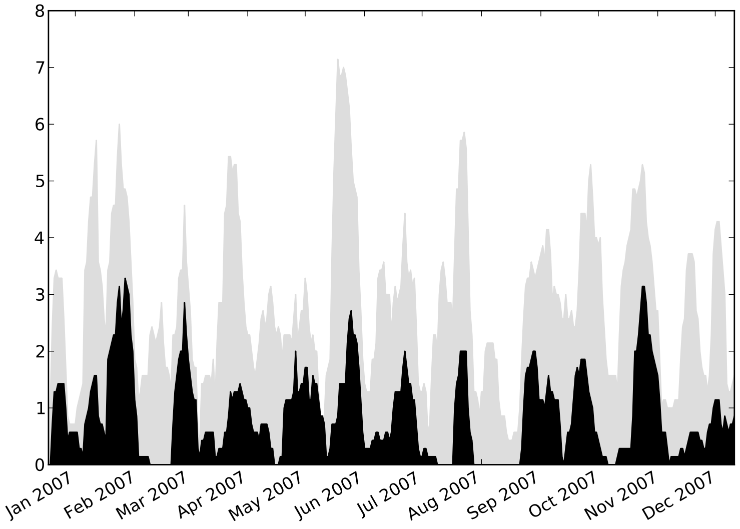


Figure 5.5: Map of selected users for full coverage at *thresh* = 80 and $m = 3$.

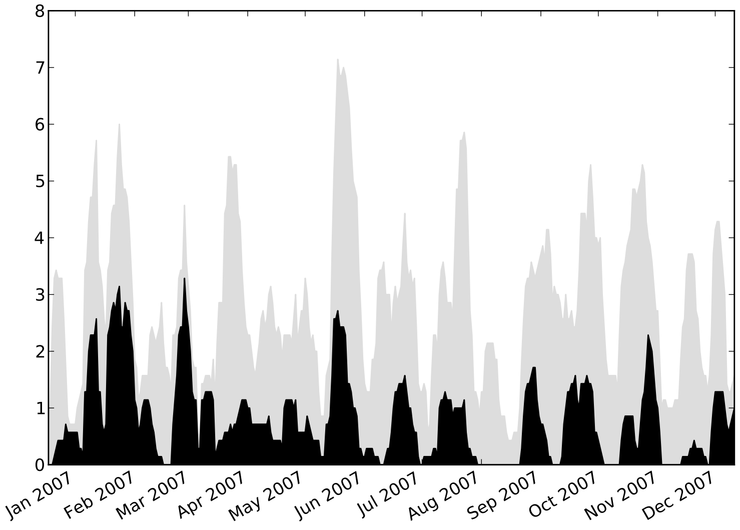
utilized as an indicator of important threads by the EIN manager and is regularly consulted on emerging epidemiological and clinical issues.

5.3.3 Analysis of Selected Threads

Using the procedure mentioned in the introduction, the set S of selected users can be used to mark threads as "to be monitored." Ideally, we would like the number of "to be monitored" threads small relative to the total number of threads. Table 5.4(a) shows the number of threads and posts observed in 2007 and the number of threads that would have been marked and number of posts that would have been read, had this procedure been in place then. For both $m = 2$ and $m = 3$, the number of marked threads are about a fourth of the total and the number of posts are about a third of the total. Per-day traffic levels, with and without use of the monitoring procedure, are given in figure 5.6 for full coverage and $m = 2$ (figure 5.6(a)) and $m = 3$ (figure 5.6(b)) for the 2007 year. Table 5.4(b) shows, for each value of m, the mailing at which important threads would have been marked using this procedure. Note that our measurement only counts posts as needing to be reads *after* a thread has been marked as important, explaining why for $m = 3$ the number of posts read decreases. Together the two tables show that as we go from $m = 2$ to $m = 3$ the cost of monitoring falls (40 threads to 38 threads, 144 posts to 140 posts) accompanied by a delay in marking a few threads (5). 5.6(b) respectively,



(a)



(b)

Figure 5.6: Plots of per-day post traffic on the EIN, averaged over seven days, in 2007. Results are from running **Greedy** with *thresh* = 80, full coverage, and (a) $m = 2$ and (b) $m = 3$. Gray levels show the total number of posts to the EIN. Black levels indicate the number of posts to marked threads only.

Table 5.4: EIN traffic statistics for the year 2007 for full coverage at $thresh = 80$.

(a)

m		Total	Marked	Imp.	Unimp.
2	Threads	229	54	14	40
	Posts	1015	314	170	144
3	Threads	229	52	14	38
	Posts	1015	289	149	140

(b)

		Mailing Marked	
m	1st	2nd	3rd
2	2	12	
3	2	7	5

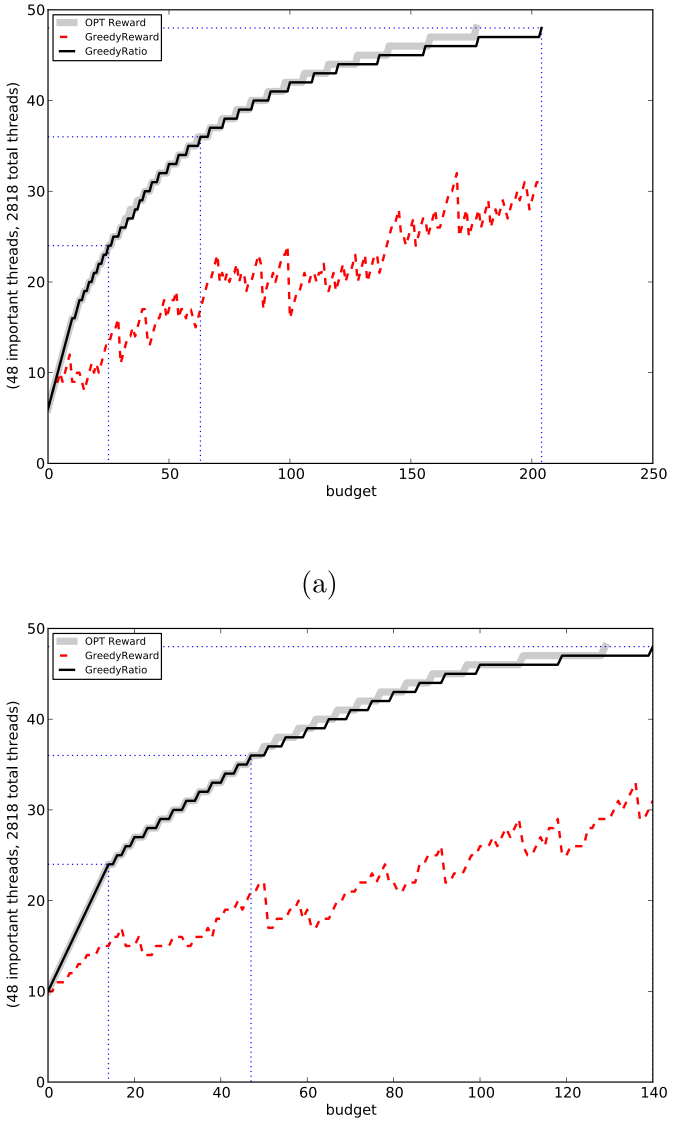
5.3.4 An Upper-Bound to **Greedy**

An instance of BMOC, $\{G, B\}$, with user-thread graph $G = \{U, T = T^+ \cup T^-, E\}$ and budget B, can be formulated as an integer linear program (IP) in the following way. Let variable u_i correspond to user $i \in U$, variable g_j correspond to important thread $j \in T^+$, and variable b_k correspond to unimportant thread $k \in T'$. To simplify and abuse a bit of notation, let $i \in j$ denote all users i such that i participate in thread j. Solving the following IP will give a solution $S = \{i|u_i = 1\}$ to BMOC.

$$
\begin{aligned}
& \max \sum_{j} g_j \\
\text{subject to} \quad & g_j \leq \sum_{i \in j} u_i & \forall i \\
& u_i \leq b_k & \forall i \in k, \forall k \\
& \sum_{k} b_k \leq B \\
& u_i, g_j, b_k \in \{0, 1\} & \forall i, \forall j, \forall k
\end{aligned}
$$

If we relax the integer constraints on u_i, g_j, b_k we get a linear program (LP) whose solution is an upper bound to the BMOC problem. Figures 5.7(a) and 5.7(b) compare the results of our greedy algorithm with the upper bound obtained by solving the LP relaxation of the above IP. Since solutions to the LP can be fractional, the objective function solutions to the LP can also be fractional, but since the solution to the BMOC problem cannot be fractional, we take the floor for the objective value given by the LP. From the figures we see the solution returned by **Greedy** achieves the upper bound except in a few cases where it is only slightly worse.

While BMOC is difficult to approximate in general, our experiments show that **Greedy** performs near-optimal on the EIN data. This can only partly be explained by overlap condition mentioned earlier. Even if we assume the overlap condition holds in our data, these plots suggest that **Greedy** produces solutions for instances of BMOC based on our data that greatly exceed a $(1-1/e)$-approximation guarantee.

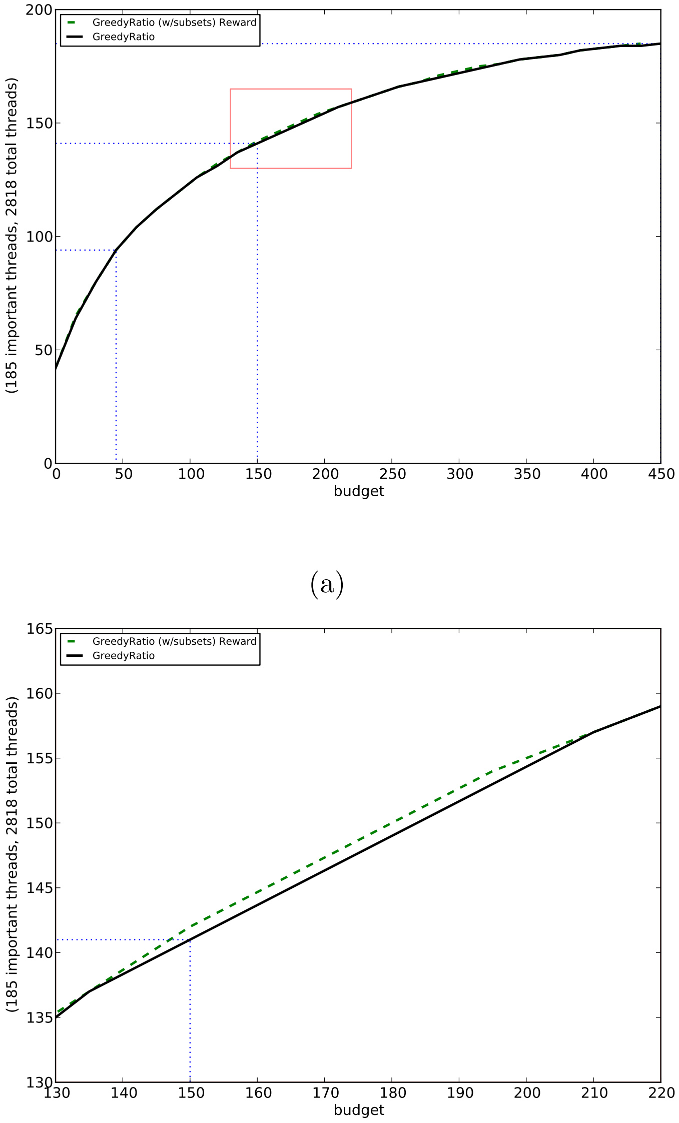


(a)

(b)

Figure 5.7: Plots comparing the performance of **Greedy** with a upperbound obtained by solving the LP relaxation of BMOC for *thresh* = 70 with (a) $m = 2$ and (b) $m = 3$. The x-axis is the budget and the y-axis is the reward. The black line shows the reward produced by the **Greedy** and the shaded line shows the upper bound obtained using the LP.

A small improvement is obtained by enhancing **Greedy** with a small "look-ahead." That is, at each step we can consider adding a subset of users, such as pairs or triples, rather only considering single users. Figure 5.8 shows the same plots as in section5.3.1 with the additional results found by modifying the **GreedyRatio** algorithm to consider pairs of users $u, u^{'} \in U$ at each iteration. While it doesn't make a significant improvement overall, the improvements are noticeable. With a very minor modification to the algorithm we find a better solution, at the cost of running time. We could improve this solution further by considering triples, or larger subsets, of users.



(a)

(b)

Figure 5.8: Plots showing the (small) improvement obtained by enhancing **Greedy** with "look-ahead;" all subsets of size at most 2 are considered as candidates in each iteration. Here $thresh = 80$ and $m = 2$. The x-axis is the budget and the y-axis is the reward. The black line shows the reward produced by **Greedy** without "look-ahead" and the dashed line shows the improvement due to "look-ahead." (a) shows the full plot and (b) shows the zoomed portion given by the box in (a).

CHAPTER 6
CONCLUSIONS AND FUTURE WORK

In the preceding chapters we have addressed a number of important practical concerns of applying social networks in epidemiology. All our research is threaded by the use of social networks to model possible disease transmitting interactions between healthcare workers in a hospital and online interactions between infectious disease experts on a mailing list. Even though our networks are generated from specific data, our results can easily extend to social networks that arise elsewhere. We conclude with a brief summary of results from each chapter and related work.

6.1 Generating HCW Contact Networks

We have first addressed the problem of using fine-grained spatiotemporal data to infer contact networks. By using our method on login data from an Electronic Medical Record (EMR) system we have constructed complex networks that approximate the contact patterns of healthcare workers (HCWs) in the University of Iowa Hospitals and Clinics. Further, we have shown evidence that these *HCW contact networks* are similar to other real-world social networks.

There are a number of alternate ways, briefly discussed in Section 2.2, we can model the EMR login data from the University of Iowa Hospitals and clinics. For example, we can model this data as bipartite graph of *people* and *locations* where edges connect people to the locations where they access the EMR system. These graphs could be used to address optimization problems related to resource placement or architecture design.

We have also not addressed the issue of validating of the HCW contact networks we construct. Recent work at the University of Iowa is exploring the use of wireless "badges", worn by HCWs and patients, to detect and record the proximity of individuals [31] and monitor hand-hygiene events [96]. It is possible that this data can be used to confirm the quality of our EMR login data and the HCW contact networks we generate.

6.2 Random Graph Models of Contact Networks

Using the networks we generate we have uncovered a number of caveats for using a particular random graph model for the study of disease diffusion. Simulation results suggest that random graph models, which only preserve vertex local properties such as average degree and degree sequence, may ignore structural properties important to accurate modeling of disease diffusion. More specifically, depending on the transmissibility of the disease being simulated, these random graph models tend to overestimate or underestimate the rate and number of infections. This difference may be due to the fact that these random graph models preserve one or more feature of the original graph and then allow mixing of all other aspects. Comparing mean and median values over multiple simulations suggests that, compared to Erdös-Rényi (ER) random graphs, outbreaks are less common on our HCW contact networks but, when they do occur, they are much more extensive. In addition, we have also introduced a new *spatial-clustering* model for the generation of random graphs with given clustering coefficient that, unlike other recently proposed models, generate graphs that have the exact degree sequence and similar clustering as our HCW contact networks. This suggests that the clustering we see in the HCW contact networks may be partially due to the spatial behaviour of individuals encoded in login data that we use to generate these graphs. Based on simulations on random graphs with specified clustering and degree sequence, clustering appears to play a minimal role on the outcome of disease diffusion. On the other hand, random graph models that preserve assortativity (degree correlations between the endpoint vertices of edges) may be the best representatives of our HCW contact networks.

While we have compared our HCW contact networks to graph models which preserve clustering and assortativity independently, we have not yet tested random graph models that preserve *both* clustering and assortativity. It is possible that random graph models that preserve both assortativity and clustering coefficient, in addition to degree sequence, may be the most accurate models for our HCW contact networks. Extending the Spatial-Clustering random graph model to incorporate both clustering and assortativity seems simple in theory, but as with the clustering models presented by Newman [87] and Bansal et al. [8], there may be practical concerns in actually generating these graphs.

6.3 Vaccination Policies

Considerable effort has been spent on understanding the problem of finding optimal vaccination policies. Our results show that the optimal set of individuals to vaccinate, in order to reduce disease diffusion, is highly dependent on the disease being controlled and structure of the underlying contact networks. Thus, when deciding on control policies based on analytical calculations and experiments, assumptions about the underlying graph could lead to poor policy decisions. However, for our HCW contact networks, results suggest that the best solution to the vaccination problem *is* to pick the most well-connected individuals. This behavior is likely a result of the high assortativity that we see in these networks and thus may be an intrinsic property of highly assortative graphs arising elsewhere. Moreover, the policy of picking the most "mobile" HCWs appears to be a fairly good solution and may be more easily implemented in practice.

In general, there are a number of practical problems relating to quarantine, isolation, and vaccination that can be modeled as optimization problems. One of our original goals was to leverage properties of the HCW contact networks to provide improved approximation guarantees to these problems. Based on the work presented here, we believe there may be ways to improve approximation guarantees for optimization problems on our HCW contact networks.

6.4 Budgeted Maximum Coverage

Finally, we have introduced the *budgeted maximum coverage with overlapping costs* (BMOC) problem to model the problem of finding "key" users of a mailing list for the purposes of disease surveillance. By leveraging possible properties of the underlying social network we are able to show that a simple approach can give near-optimal results. These results can likely apply to other social networks.

The problem of identifying which threads on the EIN are “important” is still open. Recall that we assume important threads are those that, roughly speaking, have a high number of unique participants, have longevity, and have lots of traffic (postings). It would be interesting to validate our assumption is valid by analyzing the content of these postings using semantic analysis approaches.

The EIN also presents some other interesting problems with different applications. For example, the EIN regularly sends out targeted surveys so there may be interest in finding key individuals based on aspects of their EIN participation (geography, specialization, etc.) for matters of surveillance. There is also interest in determining the “social network” of EIN user relationships which would, along with data on EIN mailing list activity, expose some interesting problems on network diffusion models.

REFERENCES

[1] EIN: H1N1 and HCW Reassignment Questions. `http://news.idsociety.org/idsa/issues/2009-09-01/3.html`, Sep 2009.

[2] EIN: H1N1 Vaccine and Guillain-Barré Syndrome. `http://news.idsociety.org/idsa/issues/2009-08-01/4.html`, Aug 2009.

[3] EIN: Treatment Options for H1N1 Pneumonia and Antiviral Use in Infants. `http://news.idsociety.org/idsa/issues/2009-07-31/5.html`, Jul 2009.

[4] Y.-Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong. Analysis of topological characteristics of huge online social networking services. In *WWW*, pages 835–844, 2007.

[5] R. Albert and A. Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[6] R. M. Anderson and R. M. May. *Infectious diseases of humans: dynamics and control*. Oxford University Press, USA, 1992.

[7] J. Aspnes, K. Chang, and A. Yampolskiy. Inoculation strategies for victims of viruses and the sum-of-squares partition problem. *Journal of Computer and System Sciences*, 72(6):1077–1093, 2006.

[8] S. Bansal, S. Khandelwal, and L. Meyers. Exploring biological network structure with clustered random networks. *BMC bioinformatics*, 10(1):405, 2009.

[9] A. Barabasi, H. Jeong, Z. Neda, E. Ravasz, A. Schubert, and T. Vicsek. Evolution of the social network of scientific collaborations. *Physica A*, 311:590–614, 2002.

[10] A. Barabási and E. Bonabeau. Scale-free networks. *Scientific American*, 288(5):50–9, 2003.

[11] A.-L. Barabasi. *Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life*. Plume Books, April 2003.

[12] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[13] A.-L. Barabási, R. Albert, and H. Jeong. Scale-free characteristics of random networks: the topology of the world-wide web. *Physica A: Statistical Mechanics and its Applications*, 281(1-4):69–77, 2000.

[14] E. Bender and E. Canfield. The asymptotic number of labeled graphs with given degree sequences. *Journal of Combinatorial Theory, Series A*, 24(3):296–307, 1978.

[15] A. Bhaskara, M. Charikar, E. Chlamtac, U. Feige, and A. Vijayaraghavan. Detecting high log-densities: an o (n) approximation for densest k-subgraph. *Proceedings of the 42nd ACM Symposium on Theory Of Computing*, pages 201–210, 2010.

[16] B. Bollobás, S. Janson, and O. Riordan. Sparse random graphs with clustering. *Random Structures & Algorithms*, 38(3):269–323, 2011.

[17] A. Bonato. A survey of properties and models of on-line social networks. In *Proc. of the 5th International Conference on Mathematical and Computational Models, ICMCM*, 2009.

[18] M. Brisson and W. J. Edmunds. Economic evaluation of vaccination programs: The impact of Herd-Immunity. *Medical Decision Making*, 23(1):76–82, 2003.

[19] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. *Computer networks*, 33(1-6):309–320, 2000.

[20] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 2011.

[21] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a submodular set function subject to a matroid constraint (extended abstract). In *IPCO '07: Proceedings of the 12th international conference on Integer Programming and Combinatorial Optimization*, pages 182–196, Berlin, Heidelberg, 2007. Springer-Verlag.

[22] D. Callaway, J. Hopcroft, J. Kleinberg, M. Newman, and S. Strogatz. Are randomly grown graphs really random? *Physical Review E*, 64(4):041902, 2001.

[23] F. Carrat, E. Vergu, N. M. Ferguson, M. Lemaitre, S. Cauchemez, S. Leach, and A.-J. Valleron. Time lines of infection and disease in human influenza: A review of volunteer challenge studies. *American Journal of Epidemiology*, 167(7), 2008.

[24] Centers for Disease Control. CDC - Adenoviruses. http://www.cdc.gov/ncidod/dvrd/revb/respiratory/eadfeat.htm, Jan 2005.

[25] C. Chekuri. Personal Communication, 2009.

[26] P. Chen, M. David, and D. Kempe. Better vaccination strategies for better people. In *Proceedings of the 11th ACM conference on Electronic commerce*, pages 179–188. ACM, 2010.

[27] F. Chung and L. Lu. Connected components in random graphs with given expected degree sequences. *Annals of Combinatorics*, 6(2):125–145, 2002.

[28] A. Clauset, C. Rohilla Shalizi, and M. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009.

[29] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70(6):066111, Dec 2004.

[30] G. Csárdi and T. Nepusz. The igraph software package for complex network research. *InterJournal Complex Systems*, 1695(1695), 2006.

[31] D. Curtis, S. Pemmaraju, L. Polgreen, P. Polgreen, and A. Segre. Contact patterns for HCWs: Not everyone is the "average". In *21st Annual Scientific Meeting of the Society for Healthcare Epidemiology of America*, 2011.

[32] D. Curtis, S. Pemmaraju, and P. Polgreen. Budgeted maximum coverage with overlapping costs: Monitoring the emerging infections network. *Workshop on Algorithm Engineering and Experiments (ALENEX)*, Jan 2010.

[33] D. Easley and J. Kleinberg. *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge Univ Pr, 2010.

[34] K. El-Arini, G. Veda, D. Shahaf, and C. Guestrin. Turning down the noise in the blogosphere. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, pages 289–298, New York, NY, USA, 2009. ACM.

[35] P. Erdős and A. Rényi. On random graphs I. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

[36] S. Eubank, H. Guclu, V. Kumar, M. Marathe, A. Srinivasan, Z. Toroczkai, and N. Wang. Modelling disease outbreaks in realistic urban social networks. *Nature*, 429(6988):180–184, 2004.

[37] S. Eubank, V. Kumar, M. Marathe, A. Srinivasan, and N. Wang. Structure of social contact networks and their impact on epidemics. *Discrete Methods in Epidemiology*, page 181, 2007.

[38] S. Eubank, V. Kumar, M. Marathe, A. Srinivasan, and N. Wang. Structural and algorithmic aspects of massive social networks. *Proceedings of the Fifteenth annual ACM-SIAM Symposium on Discrete Algorithms*, pages 718–727, 2004.

[39] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 251–262, 1999.

[40] U. Feige, G. Kortsarz, and D. Peleg. The dense k-subgraph problem. *Algorithmica*, 29:2001, 1999.

[41] M. Ferrari, S. Bansal, L. Meyers, and O. Bjornstad. Network frailty and the geometry of herd immunity. *Proceedings of the Royal Society B: Biological Sciences*, Jan 2006.

[42] R. Forsythe, F. Nelson, G. Neumann, and J. Wright. The explanation and prediction of presidential elections: a market alternative to polls. *Economics Working Paper*, pages 90–11, 1990.

[43] R. Gandhi, S. Khuller, and A. Srinivasan. Approximation algorithms for partial covering problems. *Jornal of Algorithms*, 38(4):597–608, Jan 2004.

[44] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. WH Freeman & Co. New York, NY, USA, 1979.

[45] J. Ginsberg, M. Mohebbi, R. Patel, L. Brammer, M. Smolinski, and L. Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457(7232):1012–4, 2008.

[46] M. Girvan and M. E. Newman. Community structure in social and biological networks. *PNAS*, 99(12):7821–7826, June 2002.

[47] V. Gómez, A. Kaltenbrunner, and V. López. Statistical analysis of the social network and discussion threads in slashdot. In *Proceeding of the 17th International Conference on World Wide Web*, pages 645–654, Beijing, China, 2008. ACM.

[48] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan. Approximation analysis of influence spread in social networks. *Arxiv Preprint arXiv:1008.2005*, Aug 2010.

[49] M. Hajiaghayi, K. Jain, K. Konwar, L. Lau, I. Mandoiu, A. Russell, A. Shvartsman, and V. Vazirani. The minimum k-colored subgraph problem in haplotyping and dna primer selection. *Proceedings of International Workshop of Bioinformatics Research and Applications (IWBRA)*, pages 758–766, 2006.

[50] H. W. Hethcote. The mathematics of infectious diseases. *SIAM Review*, 42(4):599–653, 2000.

[51] P. Holme and B. Kim. Growing scale-free networks with tunable clustering. *Physical Review E*, 65(2):26107, 2002.

[52] W. Jarvis. Selected aspects of the socioeconomic impact of nosocomial infections: morbidity, mortality, cost, and prevention. *Infection Control and Hospital Epidemiology*, 17(8):552–557, 1996.

[53] J. A. Jernigan, M. G. Titus, D. H. M. Groschel, S. I. Getchell-White, and B. M. Farr. Effectiveness of Contact Isolation during a Hospital Outbreak of Methicillin resistant Staphylococcus aureus. *American Journal of Epidemiology*, 143(5):496–504, 1996.

[54] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, Aug 2003.

[55] S. Khuller, A. Moss, and J. Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999.

[56] J. M. Kleinberg. The convergence of social and technological networks. *Communications of the ACM*, 51(11):66–72, 2008.

[57] G. Kossinets, J. M. Kleinberg, and D. J. Watts. The structure of information pathways in a social communication network. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, pages 435–443. ACM, 2008.

[58] G. Kossinets and D. J. Watts. Empirical Analysis of an Evolving Social Network. *Science*, 311(5757):88–90, 2006.

[59] A. Kulik, H. Shachnai, and T. Tamir. Maximizing submodular set functions subject to multiple linear constraints. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 545–554. Society for Industrial and Applied Mathematics, 2009.

[60] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. Stochastic models for the web graph. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 57–65. IEEE, 2000.

[61] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. Random graph models for the web graph. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 57–65, 2000.

[62] L. Kyne, M. B. Hamel, R. Polavaram, and P. Kelly. Health Care Costs and Mortality Associated with Nosocomial Diarrhea Due to Clostridium difficile. *Clinical Infectious Diseases*, 02215:1–8, 2002.

[63] J. Lee, V. Mirrokni, V. Nagarajan, and M. Sviridenko. Non-monotone submodular maximization under matroid and knapsack constraints. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 323–332. ACM, 2009.

[64] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani. Kronecker graphs: An approach to modeling networks. *J. Mach. Learn. Res.*, 11:985–1042, March 2010.

[65] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. S. Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429, 2007.

[66] A. Mas and E. Moretti. Peers at work. *American Economic Review*, 99(1):112–145, 2009.

[67] M. Mello and T. Brennan. Legal concerns and the influenza vaccine shortage. *JAMA: the journal of the American Medical Association*, 2005.

[68] L. Meyers, M. Newman, and B. Pourbohloul. Predicting epidemics on directed contact networks. *Journal of Theoretical Biology*, 240(3):400–418, Jun 2006.

[69] L. Meyers. Contact network epidemiology: Bond percolation applied to infectious disease prediction and control. *Bulletin: American Mathematical Society*, 44(1):63, 2007.

[70] L. Meyers, M. Newman, M. Martin, and S. Schrag. Applying network theory to epidemics: control measures for mycoplasma pneumoniae outbreaks. *Emerging Infectious Diseases*, 9(2):204–210, 2003.

[71] L. A. Meyers, B. Pourbohloul, M. Newman, D. M. Skowronski, and R. C. Brunham. Network theory and SARS: predicting outbreak diversity. *Journal of Theoretical Biology*, 232(1):71–81, January 2005.

[72] S. Milgram. The small world problem. *Psychology Today*, 2(1):60–67, 1967.

[73] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 29–42, New York, NY, USA, 2007. ACM.

[74] M. Mitzenmacher. A brief history of generative models for power law and lognormal distributions. *Internet Mathematics*, 1(2), 2003.

[75] M. Molloy and B. Reed. A critical point for random graphs with a given degree sequence. *Random structures and algorithms*, 6(2-3):161–180, 1995.

[76] M. Molloy and B. Reed. The size of the giant component of a random graph with a given degree sequence. *Combinatorics, Probability, and Computing*, 7:295–306, 1998.

[77] P. Munz, I. Hudea, J. Imad, and R. Smith. When zombies attack!: mathematical modelling of an outbreak of zombie infection. *Infectious Disease Modelling Research Progress, ed. JM Tchuenche and C. Chiyaka, Hauppauge NY: Nova Science Publishers*, pages 133–150, 2009.

[78] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of approximations for maximizing submodular set functions-1. *Mathematical Programming*, 14(1):265–294, 1978.

[79] M. Newman. *Networks: An Introduction*. Oxford Univ Pr, 2010.

[80] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69, 2004.

[81] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, February 2004.

[82] M. Newman. Assortative mixing in networks. *Physical Review Letters*, 89(20):208701, 2002.

[83] M. Newman. Random graphs as models of networks. *Arxiv preprint cond-mat/0202208*, 2002.

[84] M. Newman. Spread of epidemic disease on networks. *Physical Review E*, 66(1):016128, 2002.

[85] M. Newman. Mixing patterns in networks. *Physical Review E*, 67(2):026126, 2003.

[86] M. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3):036104, 2006.

[87] M. Newman. Random graphs with clustering. *Physical Review Letters*, 103(5):58701, 2009.

[88] M. Newman and M. Girvan. Mixing patterns and community structure in networks. *Statistical Mechanics of Complex Networks*, pages 66–87, 2003.

[89] M. Newman, S. Strogatz, and D. Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64(2):26118, 2001.

[90] N. Parikh. Generating random graphs with tunable clustering coefficient. Master's thesis, Virginia Tech, Blacksburg, VA, March 2011.

[91] R. Pastor-Satorras and A. Vespignani. Epidemic dynamics and endemic states in complex networks. *Physical Review E*, 63(6):066117, 2001.

[92] A. Perisic and C. T. Bauch. Social contact networks and disease eradicability under voluntary vaccination. *PLoS Computational Biology*, 5(2), February 2009. PMID: 19197342 PMCID: 2625434.

[93] P. PM, C. Y, P. DM, and N. FD. Using internet searches for influenza surveillance. *Clinical Infectious Diseases*, 47(11):1443–8, 2008.

[94] P. Polgreen. Personal Communication, 2009.

[95] P. Polgreen, Z. Chen, A. Segre, M. Harris, M. Pentella, and G. Rushton. Optimizing influenza sentinel surveillance at the state level. *American Journal of Epidemiology*, 170(10):1300, 2009.

[96] P. Polgreen, C. Hlady, M. Severson, A. Segre, and T. Herman. Method for automated monitoring of hand hygiene adherence without radio-frequency identification. *Infection Control and Hospital Epidemiology*, 31:1294–1297, 2010.

[97] P. Polgreen, F. Nelson, G. Neumann, and R. Weinstein. Use of prediction markets to forecast infectious disease activity. *Clinical Infectious Diseases*, 44(2):272, 2007.

[98] P. Polgreen, T. Tassier, S. Pemmaraju, and A. Segre. Using social networks to prioritize vaccination strategies for healthcare workers. *Infection Control and Hospital Epidemiology*, 31(9), 2010.

[99] A. Sala, L. Cao, C. Wilson, R. Zablit, H. Zheng, and B. Zhao. Measurement-calibrated graph models for social network experiments. In *Proceedings of the 19th international conference on World wide web*, pages 861–870. ACM, 2010.

[100] C. D. Salgado, E. T. Giannetta, F. G. Hayden, and B. M. Farr. Preventing nosocomial influenza by improving the vaccine acceptance rate of clinicians *. *Infection Control and Hospital Epidemiology*, 25(11):923–928, November 2004.

[101] A. Signori. Monitoring swine flu using twitter. `http://www.cs.uiowa.edu/~asignori/projects/twitter-monitor-swine-flu/`, Sept 2009.

[102] S. Soffer and A. Vázquez. Network clustering coefficient without degree-correlation biases. *Physical Review E*, 71(5):057101, 2005.

[103] B. Starfield. Is US Health Really the Best in the World? *JAMA: The Journal of the American Medical Association*, 284(4):483–485, July 2000.

[104] M. Stoer and F. Wagner. A simple min-cut algorithm. *Journal of the ACM (JACM)*, 44(4):585–591, 1997.

[105] T. Ueno and N. Masuda. Controlling nosocomial infection based on structure of hospital social networks. *Journal of Theoretical Biology*, 254(3):655–666, Oct 2008.

[106] A. Vázquez. Growing network with local rules: Preferential attachment, clustering hierarchy, and degree correlations. *Physical Review E*, 67(5):056104, 2003.

[107] E. Volz. Random networks with tunable degree distribution and clustering. *Physical Review E*, 70(5):056115, 2004.

[108] A. Vullikanti. Exact stochastic simulation of epidemics on complex social contact networks. Personal Communication.

[109] D. Watts and S. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.

[110] R. P. Wenzel and M. B. Edmond. The impact of hospital-acquired bloodstream infections. *Emerging Infectious Diseases*, 7(2):174–177, 2001. PMID: 11294700 PMCID: 2631709.

[111] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.