MODELING AND SIMULATING THE PROPAGATION OF INFECTIOUS DISEASES USING COMPLEX NETWORKS

A Thesis
Presented to
The Academic Faculty

by

Rick Quax

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in
Computer Science

College of Computing
Georgia Institute of Technology
August 1, 2008

MODELING AND SIMULATING THE PROPAGATION OF INFECTIOUS DISEASES USING COMPLEX NETWORKS

Approved by:

David Bader, Ph.D., Committee Chair
Computational Science and Engineering
Georgia Institute of Technology

Peter Sloot, Ph.D., Advisor
Section of Computational Science
University of Amsterdam

Richard Vuduc, Ph.D.
Computational Science and Engineering
Georgia Institute of Technology

Date Approved: July 7, 2008

ACKNOWLEDGEMENTS

I would like to thank my thesis advisors, who have guided me in the presentation of my thesis, and Shan Mei, with whom I have had many insightful discussions.

Contents

List of Tables

List of Figures

SUMMARY

For explanation and prediction of the evolution of infectious diseases in populations, researchers often use simplified mathematical models for simulation. We believe that the results from these models are often questionable when the epidemic dynamics becomes more complex, and that developing more realistic models is intractable.

In this dissertation we propose to simulate infectious disease propagation using dynamic and complex networks. We present the Simulator of Epidemic Evolution using Complex Networks (SEECN), an expressive and high-performance framework that combines algorithms for graph generation and various operators for modeling temporal dynamics. For graph generation we use the Kronecker algorithm, derive its underlying statistical structure and exploit it for a variety of purposes. Then the epidemic is evolved over the network by simulating the dynamics of the population and the epidemic simultaneously, where each type of dynamics is performed by a separate operator. All dynamics operators can be fully and independently parameterized, facilitating incremental model development and enabling different influences to be toggled for differential analysis.

As a prototype, we simulate two relatively complex models for the HIV epidemic and find a remarkable fit to reported data for AIDS incidence and prevalence. Our most important conclusion is that the mere dynamics of the HIV epidemic is sufficient to produce rather complex trends in the incidence and prevalence statistics, e.g. without the introduction of particularly effective treatments at specific times. We show that this invalidates assumptions and conclusions made previously in the literature, and argue that simulations used for explanation and prediction of trends

should incorporate more realistic models for both the population and the epidemic than is currently done. In addition, we substantiate a previously predicted paradox that the availability of Highly Active Anti-Retroviral Treatment likely causes an increased HIV incidence.

Chapter I

INTRODUCTION

1.1 The thesis

Our thesis is that realistic simulations of epidemics over population networks are more easily and more adequately performed using annotated complex networks and independent dynamics operators. Traditional mathematical models are sufficient for relatively simple simulations but quickly become intractable as more influences and interactions are added. In contrast, using an explicit representation of a population network annotated with variables, where an epidemic is evolved through the application of separate operators, models can easily be extended in an incremental way, be specified in arbitrary detail and different effects can be toggled for differential analysis. We also claim that this method can be practical in terms of running time.

1.2 Problem statement

Understanding patterns of prevalence and incidence of infectious diseases is a major challenge for various reasons. Static population networks consist of heterogeneous agents in multiple respects, and are characterized by distinctive global properties such as power-law degree distribution, community-structure and small diameter. In addition, sociological processes change the network over time but depend on the epidemic status, and conversely the propagation of an epidemic depends on both the static and dynamic properties of the population network.

At present, most simulations of epidemics are performed using mathematical models of complex real-world dynamics with only a handful of parameters. Popular methods are coupled differential equations and discrete Markov models. However,

such approaches have significant drawbacks: the model complexity quickly becomes intractable for more realistic simulations, incrementally extending existing models is non-trivial and simplifying assumptions may have unexpected but dramatic effects on the simulation results. We argue that these models are adequate in limited cases, but that different methods should be used for more detailed and realistic knowledge discovery.

1.3 Using complex networks

1.3.1 Overview

We propose to explicitly simulate the dynamics of virus propagation using annotated complex networks. Firstly, a graph representative of the target population is generated where the nodes represent agents and the edges represent possible infection pathways in a time step. A number of operators is then applied to the graph to model temporal epidemiological and sociological dynamics of the population. Nodes and edges are annotated with static and dynamic variables, and operators are parameterized independent of each other. This enables the incremental development of models of an epidemic, toggling of different influences for differential analysis and allows for models of arbitrary complexity. For instance, an agent's infection stage and gender could influence its social behavior, infectivity, connectivity and life expectancy; at the same time, an agent's behavior and connectivity could influence the probability of becoming infected and infecting someone else.

1.3.2 The SEECN framework

We present SEECN[1], a simulation framework that is designed to be expressive, modular and high-performance. The temporal operators support addition and deletion of nodes and edges, propagation of a disease over edges and its local progression within nodes. They can be fully parameterized by the variables of the involved entities:

[1]SEECN is available at `www.science.uva.nl/~rquax` and easily pronounced as "season".

nodes have a static 'type' and a dynamic 'status', and edges have a static 'type' and store the node variables at both sides. Time is discretized in time steps and all model parameters can be changed in each time step, for instance to reflect the introduction of a new treatment.

SEECN is comprehensively engineered for performance. It is written in C++ and parallelized with OpenMP, and both the graph generation and temporal dynamics algorithms are optimized for NUMA and cache efficiency using novel theory of Kronecker graphs. Even though the algorithms are bounded by memory-bandwidth and have inherently unpredictable access patterns, we were able to achieve a speed-up of one order of magnitude in single-processor performance and almost perfect scalability in the number of threads.

1.4 Applicability to other domains

SEECN was developed for simulating infectious diseases, but we expect that it can be used for many other applications as well. The paradigm of studying the spread of some 'virus' over a domain of nodes and edges is a general one, although we currently only generate social interaction networks which are characterized by distinctive properties such as power-law degree distribution, small diameter and community structure. Such properties and dynamics are found in e.g.:

- Information dissemination in blog communities;
- Interaction between proteins;
- Connectivity simulations of file-sharing networks;
- Robustness of autonomous system networks against failures.

1.5 Our contribution

The contribution of this dissertation is twofold. In chapter 3 we choose Kronecker graphs for modeling social networks, derive novel and complete statistical properties of the total edge count, degrees of individual nodes and edge configuration, and exploit these in a variety of applications that were impossible or impractical before. RMAT is an algorithm that has been proposed to generate Kronecker graphs efficiently, and is described in chapter 2. In particular, we show how the results enable load-balancing and cache efficiency of the RMAT algorithm; combined with a refactored graph data structure and the use of the same techniques in the dynamics operators, we expect that SEECN is practical. We also suggest how the posterior probability of a graph can be computed efficiently and how a simple mathematical model can incorporate the statistical structure of a Kronecker graph.

Combining the graph generator with modules for network dynamics, infection propagation and local virus progression, we present and validate our framework in chapter 4. In chapter 5 we then showcase its expressiveness by performing a non-trivial simulation of the HIV epidemic. Firstly, we find a surprising phenomenon that could have caused the observed drop in AIDS annual incidence that invalidates earlier assumptions and conclusions, namely that the HIV epidemic itself is sufficient to produce such complex trends in reported data, and demonstrate that the use of simplistic mathematical models is insufficient as basis for complex predictions. As a corollary we also add further evidence to support a much-debated paradoxical prediction that the introduction of treatment could be counter effective and find indeed that it likely results in higher stabilized HIV incidence.

Chapter II

PREVIOUS WORK

Here we introduce some terminology, review important graph properties and summarize previous work on graph generation and modeling and simulation of epidemics.

2.1 Graph properties

In this dissertation we define a graph (or 'network') G as a set of nodes and edges $\langle V, E \rangle$, and we define $|V| = N = 2^n$ where n is an integer. All our graphs are *undirected*, i.e. $\forall_{x,y \in V}\ (x, y) \in E \Rightarrow (y, x) \in E$; if this does not hold the graph is said to be *directed*. A graph has a *power-law distribution* when the number of nodes that has a particular degree k decreases monomially as k increases, i.e. $p_k \propto k^{-\gamma}$ where γ is the power-law exponent and typically $1 < \gamma < 3$. For increasing N, a graph follows the *densification law* if $|E| \propto N^{\alpha}$, $\alpha > 1$ where typically $1 < \alpha < 2$. In other words, the average degree increases as the graph size increases. This was recently observed to be true for various ubiquitous graphs such as social networks and the World Wide Web [16] and appears to coincide with a shrinking diameter (while it was long debated whether these networks have e.g. $\mathcal{O}(\log N)$ or $\mathcal{O}(\log \log N)$ diameter *growth*). The *diameter* of a graph is typically defined as the maximum path length of all possible shortest paths, although in practical situations *effective diameter* is often used; this states a diameter such that some considerable fraction of shortest paths is shorter than this, where the fraction is typically $\geq 90\%$. This accommodates isolated nodes and excessively rare long paths.

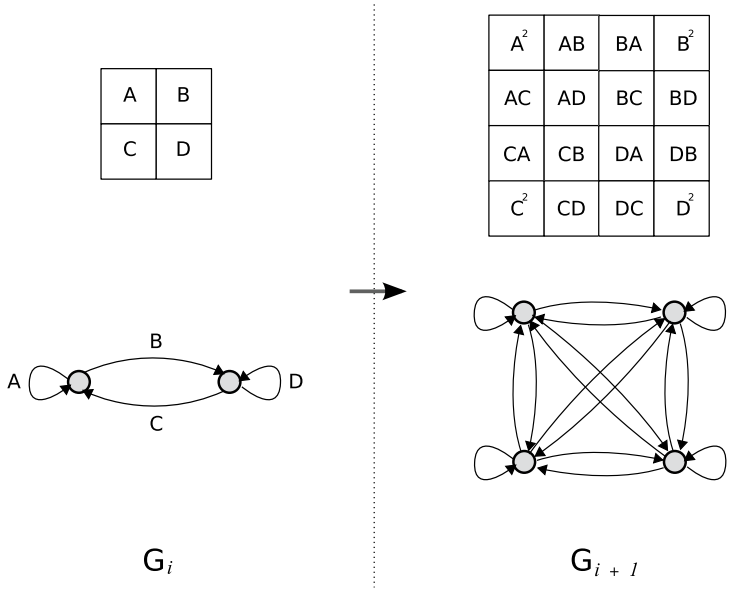


Figure 1: An illustration of a step in the Kronecker algorithm. The scalar entries of matrix G_i are multiplied with the matrix itself and replaced with the result, to yield G_{i+1}.

2.2 Kronecker and RMAT

Leskovec et al. [17], proposes to adopt the *Kronecker* matrix multiplication to generate realistic and mathematically tractable graphs. A large body of mathematical theory has already been developed for this operator, which they intended to exploit. Conceptually, the Kronecker multiplication of a $q \times q$ matrix with itself yields a $q^2 \times q^2$ matrix where each original (scalar) entry is multiplied by the entire original matrix, and replaced with the result. When a small matrix G_1 is repeatedly Kronecker-multiplied with itself, the result can be interpreted as an adjacency matrix; with the right choice of parameters, properties such as a power-law distribution and hierarchical community-structure can be obtained. See figure 1 for an illustration.

In the stochastic case, a small *initiator matrix* contains edge probabilities $0 \leq \Theta_{i,j} \leq 1$. The result $P = \Theta^t$ after t Kronecker time steps is a probabilistic adjacency matrix where each $P_{x,y}$ is a unique permutation of multiplying $\prod_{(i.j)} \Theta_{i,j}$ and denotes

the probability of edge (x, y) being present [15].[1] This initiator matrix is typically very small and often of size 2×2, so that the bit-representations of two unique ids $i, j \in [0, N)$ of two particular nodes uniquely identify such a permutation with n factors.

Most theoretic results were derived for the deterministic Kronecker procedure, which only allows $\forall_{i,j}\Theta_{i,j} \in \{0, 1\}$, but some results for the stochastic version are available as well. The most important are that the resulting graphs are statistically self-similar, have constant diameter, create a hierarchical community-structure and have a multinomial degree distribution that can be fitted to power-law distributions [17, 15, 19].

The RMAT algorithm [10] is a slight variation where the number of edges must be chosen beforehand and the initiator matrix is normalized, i.e. $\sum_{i,j} \tilde{\Theta}_{i,j} = 1$. This way, each edge can be 'dropped' into the initiator matrix and recursively 'chooses' one of the entries based on its probability, logically choosing the corresponding partition of the adjacency matrix until a 1×1 base case is reached where the edge is placed. Note that various concerns may arise; one must choose the number of edges beforehand and edges may duplicate. The RMAT algorithm is therefore not an *exact equivalent* to the original Kronecker algorithm, but for sparse graphs this method is much more efficient: a naïve Kronecker algorithm iterates over all possible edges and determines whether or not it is actually placed, resulting in $\mathcal{O}(N^2)$ performance whereas RMAT runs in $\mathcal{O}(|E|)$.

2.3 Simulating epidemics

Many generic, mathematical methods for modeling epidemics have been proposed [3, 9], most of which use either coupled differential equations, Markov models, cellular automata or some hybrid approach. An interesting example is that of modeling the

[1] Each such permutation occurs exactly once, and it does not matter which edge is assigned which permutation.

SARS virus [12], which surprised many epidemiologists with its fast and seemingly random spreading pattern. Various models allow for a better understanding of its contagiousness, influencing public health policies but also show the importance of taking the distinct social network characteristics into account.

Applied to HIV and AIDS, Baggaley et al. [6] review numerous mathematical models that have been developed and applied; interestingly, none have incorporated special properties of social interaction networks such as a power-law degree distribution, and none are probabilistic. In particular, Xiridou et al. [28, 29] perform quantitative analysis and predictions of the relative influence of steady and casual relationships on the HIV epidemic, but model the degrees of agents with only two uniform distributions. In Aalen et al. [2], a marked decline in AIDS incidence is explained by the introduction of more effective treatment based on a multi-stage Markov-model. They dismiss the possibility that this decline is caused simply by the dynamics of the epidemic itself, but provide no basis for this assumption. Further, Katz et al. [14] suggest that a small but distinctive increase in AIDS prevalence from 1996 is due to the coincident introduction of HAART, based merely on the observation that both seem to begin around the same time. They also predict that the benefits of HAART are counterbalanced of overwhelmed by an increase of risky behavior. Lastly, in Schwarcz et al. [24] a detailed discussion of temporal trends in HIV incidence and prevalence in San Francisco is presented, which is the same domain as that of Katz et al.

Using explicit graphs representations and spreading algorithms instead, Eubank et al. [13] take the approach of modeling infectious diseases computationally by building a very detailed social network based on real urban, everyday-life data, in particular the movement and interaction patterns of individuals. Their main goal was to identify vaccination strategies in case of catastrophes. A drawback is that such data is typically not available. Sloot and Ivanov [25] abstract away from using specific data

and studies the propagation of HIV over artificial networks that fit qualitative, global properties of the real network. They introduce the notion of network and epidemic operators, which we adopt in this dissertation. However, their graphs only obey the power-law degree distribution and are completely regenerated at every time step, leaving their fitting results questionable and essentially degenerating their simulator to a small set of differential equations by summing out over disease locality and node degrees. An illustration of the inaccuracy is as follows: high-degree nodes are infected relatively early in the process, and can spread the infection to many nodes. However, if high-degree nodes are reselected in subsequent time steps and are probably not infected anymore, the simulated epidemic spreads less quickly than the real epidemic. Finally, more theoretical results for various network types are presented in the literature such as [27, 22], such as epidemic threshold, the effect of network topology and spreading rate.

Chapter III

GENERATING SOCIAL INTERACTION NETWORKS

Social networks turn out to have a number of distinctive features, the most important of which are a power-law degree distribution, hierarchical community structure and a small degree [16, 7, 4, 5]. We have chosen to use the Kronecker algorithm for generating such networks. Not only has it been shown to fit to real-world networks with striking accuracy [15, 10, 17], we also believe the procedure can be adapted and generalized using the results from section 3.1. We leave this as future research.

In this chapter we present novel derivations of properties of Kronecker graphs and use it to make the RMAT algorithm both more efficient and statistically indistinguishable from the original Kronecker procedure, enabling theoretic results for both to be used interchangeably. In particular, we refactor the graph's data structure and show how load-balancing and cache-efficiency can be implemented to optimize the RMAT algorithm, and find remarkable speed-ups even though the algorithm is memory-intensive and has an irregular access pattern. We also suggest other possible applications of the theoretical results, for instance how the exact posterior probability $P(\mathcal{G} = G^n | a, b, c, d)$ of some graph $\mathcal{G}$ - i.e. the probability that a Kronecker algorithm with parameters a, b, c, d generates graph $\mathcal{G}$ - can be calculated in linear time complexity, in contrast to the superexponential time complexity suggested by Leskovec et al. [15]. This is used to verify the stability of the network operators of section 4.1.2.

First we clarify some terminology. The *Kronecker algorithm* is the naïve, $\mathcal{O}(N^2)$ generation algorithm (section 2.2) that logically computes the probability adjacency matrix P and then accepts or rejects each individual edge; a *Kronecker graph* has been generated by either the Kronecker algorithm or an equivalent. In particular, we

view the original *RMAT algorithm* as an approximation because edges may duplicate and an edge count $|E|$ must be chosen a priori as parameter to the algorithm, whereas the Kronecker algorithm automatically follows some probability mass function (pmf) (implicitly defined by its parameters). We believe this distinction is important if we intend to exploit theoretic results from the literature for either algorithm, and assume that the statistical bias from RMAT cannot be ignored. We use a 2×2 initiator matrix with parameters $\{a, b; c, d\}$ and focus on *undirected edges*, i.e. $b = c$. Undirectedness has far-reaching implications for our implementations and has never been addressed in detail in current literature by our knowledge; however, both our results and framework can be trivially adapted to the directed case.

In this chapter we derive the underlying statistical structure of Kronecker graphs in section 3.1, which we validate experimentally in section 3.1.1 and summarize in 3.1.2. Then we exploit this novel theory in 3.2 where we show how load-balancing and cache efficiency can be achieved in the RMAT algorithm, and combine with the use of a different graph data structure. As a result, we were able to achieve significant performance gains for both the single-threaded and multi-threaded implementation (section 3.2.5). Finally, we suggest a variety of additional applications of the novel theory in section 3.3.

3.1 Derivation of underlying structure of Kronecker graphs

Here we derive both the exact and approximate probability distributions for the degree of each node and the total edge count of a graph, based on the Kronecker algorithm. As a corollary we also find that nodes with the same degree pmf have the same probability of receiving an edge side, whether it is chosen as the first or the second side. Even though we assume undirected graphs in this dissertation, the results in this section and the applications thereof extend trivially to the directed case. In particular, we do not make any assumptions about the scalar probabilities

a, b, c, d, unless otherwise noted. The derivations are also generalizable to arbitrary initiator matrix sizes, but here we restrict ourselves to 2×2.

A Kronecker graph has $N = 2^n$ nodes which we can label as $0, 1, ..., N-1$ in any order: in particular, we represent a node label x with a unique string of n bits $\langle x_1, x_2, \ldots, x_n \rangle$ where $\forall_{x_i \in x}\ x_i \in \{0, 1\}$. Using this we first express how edge probabilities can be computed, how they build up to a node's degree and then identify categories of nodes with equal statistical properties, both in degree probability and edge configuration. Note that the labeling scheme is arbitrary and does not influence the resulting graph structure.

Lemma 3.1. *The probability of each distinct edge (x, y) can be computed by multiplication of a unique permutation of n factors $f_i \in \{a, b, c, d\}$.*

Proof. Each node's label can be uniquely specified with n bits, and an edge is uniquely identified by choosing an ordered pair of two node labels (x, y). In step i of the Kronecker algorithm a quarter of the adjacency matrix is replaced by the multiplication the previous graph $f_i \times G_i$ with a scalar $f_i \in \{a, b, c, d\}$, depending on which quarter is replaced.

When the node labels of the adjacency matrix are sorted, a choice of quarter simultaneously fixes the bits of x and y at position i, therefore the choice of the subsequent scalar factor or the two bits is equivalent. Note that each permutation of n bits is a valid node label. Obviously there is only one permutation of bits that will yield x (and the same for y), such that any other permutation of factors yields the probability for a different edge (x', y') and no two different edges can have the same permutation. □

In the following, $\omega_w(i)$ is defined as the number of bits with value w in the bit-representation of integer i. We further define $x|y$ as an array of size n of the pair-wise ordered concatenation of corresponding bits, the first from x and the second from y,

i.e. $x|y := \langle x_1y_1, x_2y_2, \ldots, x_ny_n\rangle$. Then, $\omega_{vw}(x|y)$ $(v, w \in \{0, 1\})$ gives the number of times that the element vw occurs in the array $x|y$.

Lemma 3.2. *The probability of edge (x, y) equals*

$$P((x, y) \in E) = a^{\omega_{00}(x|y)}b^{\omega_{01}(x|y)}c^{\omega_{10}(x|y)}d^{\omega_{11}(x|y)}.$$

Proof. From the proof of lemma 3.1 it follows that each consecutive choice of a particular factor a, b, c, or d selects two bits in the bit representations of the ordered pair (x, y), or equivalently one element of the array $x|y$. If we arbitrarily assign which bit-value is selected upon which parameter selection for both node ids, we can simply count the number of times each of the four possible combinations of bit-values occurs in $x|y$ and have exactly that many corresponding parameters in the multiplication permutation. By virtue of commutativity of scalar multiplication we can reorder the factors to obtain the expression. □

Corollary 3.3. *When one side of an edge is fixed at node x, it follows that*

$$\forall_{i,j\in[0,N)}\ \omega_0(y_i) = \omega_0(y_j) \Rightarrow P((x, y_i) \in E) = P((x, y_i) \in E).$$

In other words, when a node chooses a neighbor for an edge it is indifferent between candidates with equal bit-value counts in their label.

Lemma 3.4. *For any fixed node label x, the number of edges that have equal edge probability $P((x, y') \in E)$ for some y' is $\binom{\omega_0(x)}{\omega_{00}(x|y')}\binom{\omega_1(x)}{\omega_{10}(x|y')}$. From this set of edges, the contribution to the node x's degree follows the binomial distribution*

$$Bin\left[\binom{\omega_0(x)}{\omega_{00}(x|y')}\binom{\omega_1(x)}{\omega_{10}(x|y')}, P((x, y') \in E)\right].$$

Proof. If x is fixed, we fix the left-hand side of each element in the array $x|y$. If all edges are considered, our contiguous numbering scheme and the constraint on possible graph sizes (powers of 2) entail that any right-hand side permutation of bit-values

occurs exactly once. For a given $P((x, y') \in E)$ for some y' we can therefore calculate the element counts $\omega_{1i}(x|y)$ and $\omega_{0j}(x|y)$ independently. Due to commutativity of multiplying the resulting permutation of factors $f_i \in \{a, b, c, d\}$, labels y_i and y_j yield the exact same edge probability if $\forall_{v,w\in\{0,1\}}\ \omega_{vw}(x|y_i) = \omega_{vw}(x|y_j)$. The number of times all four possible array-element counts are equal to those for node y is $\binom{\omega_0(x)}{\omega_{00}(x|y)}\binom{\omega_1(x)}{\omega_{10}(x|y)}$. Lastly, the Kronecker algorithm considers each edge for additions independently, which yields the binomial contribution to the degree of node x. □

In the remainder we define $d_x(k)$ as the probability that the node with label x has degree k.

Theorem 3.5. *The exact pmf $d_x(k)$ for a node x is the convolution of $(\omega_0(x) + 1)(\omega_1(x) + 1)$ binomials,*

$$\sum_{w_{00}=0}^{\omega_0(x)} \sum_{w_{10}=0}^{\omega_1(x)} Bin\left[\binom{\omega_0(x)}{w_{00}}\binom{\omega_1(x)}{w_{10}}, a^{w_{00}} b^{\omega_0(x)-w_{00}} c^{w_{10}} d^{\omega_1(x)-w_{10}}\right],$$

where

$$P(Bin[N_1, p_1] + Bin[N_2, p_2] = k) := \sum_{i+j=k} Bin[i; N_1, p_1] \cdot Bin[j; N_2, p_2].$$

Proof. From the proof of lemma 3.4 it follows that two edge probabilities $P((x, y_i) \in E)$ and $P((x, y_i) \in E)$ cannot be assumed equal iff either $\omega_{00}(x|y_i) \neq \omega_{00}(x|y_j)$ or $\omega_{10}(x|y_i) \neq \omega_{10}(x|y_j)$. For all y, there are $\omega_0(x) + 1$ possible values for $\omega_{00}(x|y)$ and $\omega_1(x) + 1$ possible values for $\omega_{10}(x|y)$, leaving $(\omega_0(x) + 1)(\omega_1(x) + 1)$ groups of edges with distinct edge probabilities.

The contribution of each such group of edges follows the binomial from lemma 3.4 and the sum of all contributions yields node x's degree instance. □

Lemma 3.6. *The expected degree $E[d_x]$ of a node x is $(a + b)^{\omega_0(x)}(c + d)^{\omega_1(x)}$.*

Proof. The expected degree is the sum of the expected values of all binomials in the

convolution from theorem 3.5, and

$$\begin{aligned}
\mathrm{E}[d_x] &= \mathrm{E}\left[\sum_{w_{00}=0}^{\omega_0(x)}\sum_{w_{10}=0}^{\omega_1(x)} \mathrm{Bin}\left[\binom{\omega_0(x)}{w_{00}}\binom{\omega_1(x)}{w_{10}}, a^{w_{00}}b^{\omega_0(x)-w_{00}}c^{w_{10}}d^{\omega_1(x)-w_{10}}\right]\right], \\
\mathrm{E}[d_x] &= \sum_{w_{00}=0}^{\omega_0(x)}\sum_{w_{10}=0}^{\omega_1(x)} \mathrm{E}\left[\mathrm{Bin}\left[\binom{\omega_0(x)}{w_{00}}\binom{\omega_1(x)}{w_{10}}, a^{w_{00}}b^{\omega_0(x)-w_{00}}c^{w_{10}}d^{\omega_1(x)-w_{10}}\right]\right], \\
\mathrm{E}[d_x] &= \sum_{w_{00}=0}^{\omega_0(x)}\sum_{w_{10}=0}^{\omega_1(x)} \binom{\omega_0(x)}{w_{00}}\binom{\omega_1(x)}{w_{10}} \cdot a^{w_{00}}b^{\omega_0(x)-w_{00}}c^{w_{10}}d^{\omega_1(x)-w_{10}}, \\
\mathrm{E}[d_x] &= \sum_{w_{00}=0}^{\omega_0(x)} \left(\binom{\omega_0(x)}{w_{00}} a^{w_{00}}b^{\omega_0(x)-w_{00}} \cdot \sum_{w_{10}=0}^{\omega_1(x)} \binom{\omega_1(x)}{w_{10}} c^{w_{10}}d^{\omega_1(x)-w_{10}}\right), \\
\mathrm{E}[d_x] &= \sum_{w_{00}=0}^{\omega_0(x)} \binom{\omega_0(x)}{w_{00}} a^{w_{00}}b^{\omega_0(x)-w_{00}} \cdot (c+d)^{\omega_1(x)}, \\
\mathrm{E}[d_x] &= (a+b)^{\omega_0(x)} \cdot (c+d)^{\omega_1(x)}.
\end{aligned}$$

□

Theorem 3.7. *The exact pmf $\mathcal{E}(e) = P(|E| = e)$ of the total edge count of a Kronecker graph can be computed by the recursive convolution $D_N(e)$ where $D_x(e) = \sum_k d_x(k) + D_{x-1}(e-k)$ and $D_0(e) := d_0(e)$. The expected edge count $E[\mathcal{E}]$ for an undirected Kronecker graph is $\frac{1}{2}(a+b+c+d)^n$.*

Proof. For the exact pmf of a Kronecker graph it suffices to observe that the edge count of a graph is the sum of the degrees of all its nodes.

Because of undirectedness, the expected edge count is half the sum of the expected degrees of all nodes: summing over all expected degrees counts every edge twice. From lemma 3.6 it follows that nodes with equal bit value counts have equal expected degrees. Then,

$$
\begin{aligned}
\mathrm{E}\left[\mathcal{E}\right] &= \frac{1}{2}\sum_{w_0}^{n}(a+b)^{w_0}(c+d)^{n-w_0}, \\
\mathrm{E}\left[\mathcal{E}\right] &= \frac{1}{2}\left((a+b)+(c+d)\right)^{w_0+n-w_0}, \\
\mathrm{E}\left[\mathcal{E}\right] &= \frac{1}{2}(a+b+c+d)^{n}.
\end{aligned}
$$

□

All aforementioned convolutions are finite because negative degrees have probability zero and the maximum possible edge count is N^2, however the calculation is prohibitive for larger graph sizes. However, from theorem 3.7 we can characterize the evolution of the expected average degree $\mathrm{E}[\mathcal{E}]/N$ of a graph as a function of n, as

$$
\frac{\mathrm{E}[\mathcal{E}]}{N} = \frac{1}{2}\frac{(a+b+c+d)^n}{2^n} \propto \frac{a+b+c+d}{2} = \alpha
$$

where α is the densification parameter. Using this we postulate the following:

Postulate 3.8. The binomials from theorem 3.5 can be adequately approximated by a Poisson distribution for larger graph sizes.

Motivation 3.9. The average degree increases by only a modest factor if the graph size is doubled; for instance, in the simulations of HIV propagation in chapter 5, $\alpha \approx 1.15$. Therefore, as the sample size (i.e., the size of the probabilistic adjacency matrix) is increased, all edge probabilities decrease leaving the expected degree roughly the same compared to the factor of 4 for the increase of sample size.

Once the binomials from theorem 3.5 are approximated by a Poisson, the degree of any node and the total edge count of graph can be written as single Poisson distributions.

Theorem 3.10. *The pmf for the degree of a node with label x is well approximated by Poisson* $\left[(a+b)^{\omega_0(x)}(c+d)^{\omega_1(x)}\right]$.

Proof. This follows directly from the postulation and the fact that the sum of independent Poisson-distributed random variables follows a new Poisson distribution with the sum of the expected values as its parameter, i.e. $\sum_i \text{Poisson}\left[\lambda_i\right] \equiv \text{Poisson}\left[\sum_i \lambda_i\right]$. The parameter of the Poisson distribution is calculated as in the proof of lemma 3.6. □

Corollary 3.11. *In a Kronecker graph of size 2^n, at most $n+1$ 'categories' of nodes exist within which all nodes have the same degree pmf, as well as the same probability of receiving the remote side of an edge. (See also corollary 3.3.)*

With an appropriate choice of organizing nodes into *categories*, all nodes within a category have the same degree pmf and probability of being chosen as the remote side of an edge.

Definition 3.12. A *category* $\mathcal{C}_i$ is one of $n+1$ disjoint sets of nodes, labeled with a unique number $i \in [0, n]$ such that $\forall_{x \in \mathcal{C}_i}\ \omega_1(x) = i$.

Lastly, we present the degree pmf for the graph's total edge count.

Theorem 3.13. *The pmf for the total edge count of a Kronecker graph can be written as Poisson* $\left[\frac{1}{2}(a+b+c+d)^n\right]$.

Proof. This follows directly from the postulation and the transformation of the sum of independent Poisson distributions. The parameter is calculated as in the proof of theorem 3.7. □

3.1.1 Experimental validation

The approximations for the pmfs of node degrees and a graph's total edge count turn out to fit remarkably well. In figure 2 we show plots for the exact and approximate degree pmfs two nodes for a graph size of $2^8 = 256$ nodes. The fits are already very good for this small graph size, and we observe that the fits improve as the

graph size increases (not shown): for a graph size of $2^{10} = 1024$ the two curves are indistinguishable. The exact and approximate edge count pmfs of a graph are also indistinguishable and are omitted.

As a representation for 'low-degree' nodes we have taken the category with the second lowest expected degree, and for 'high-degree' nodes the second highest expected degree.

3.1.2 Summary of results

A Kronecker graph of size 2^n generated with parameters a, b, c, d has at most $n + 1$ different categories of nodes with equal properties. For all categories i,

$$|\mathcal{C}_i| = \binom{n}{i}, \tag{1}$$

$$\forall_{x \in \mathcal{C}_i}\ d_x(k) \approx \text{Poisson}\left[(a+b)^{n-i}(c+d)^i\right], \tag{2}$$

$$\forall_{y,y' \in \mathcal{C}_i}\ P\left((x,y) \in E\right) = P\left((x,y') \in E\right). \tag{3}$$

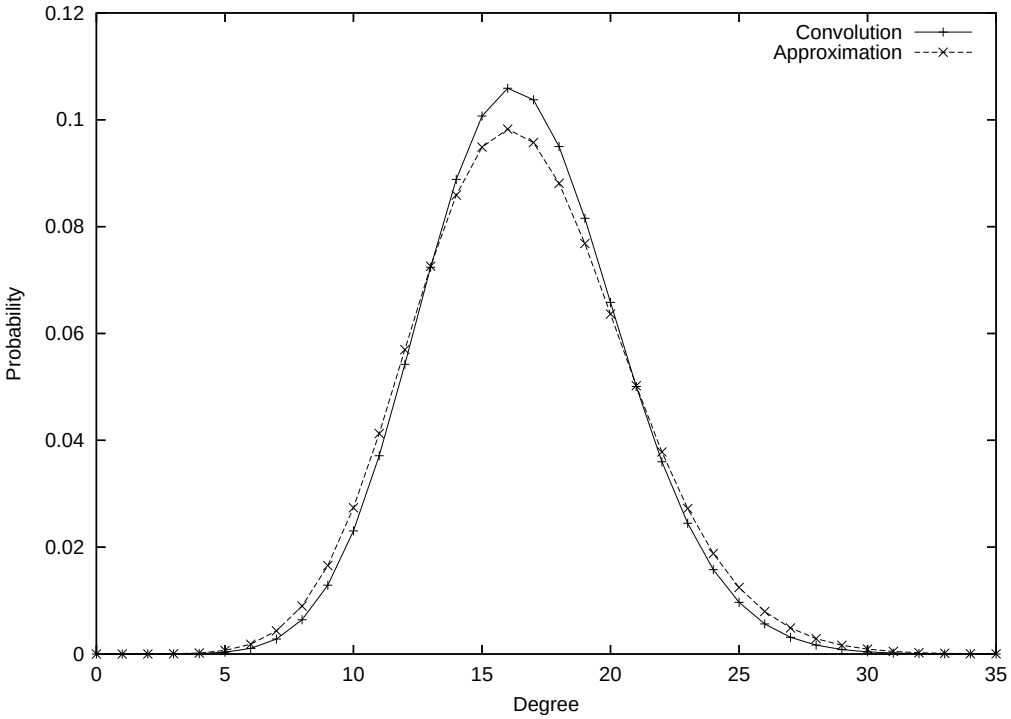
For the graph's total edge count pmf,

$$\mathcal{E}(e) \approx \text{Poisson}\left[\frac{1}{2}(a+b+c+d)^n\right] \tag{4}$$

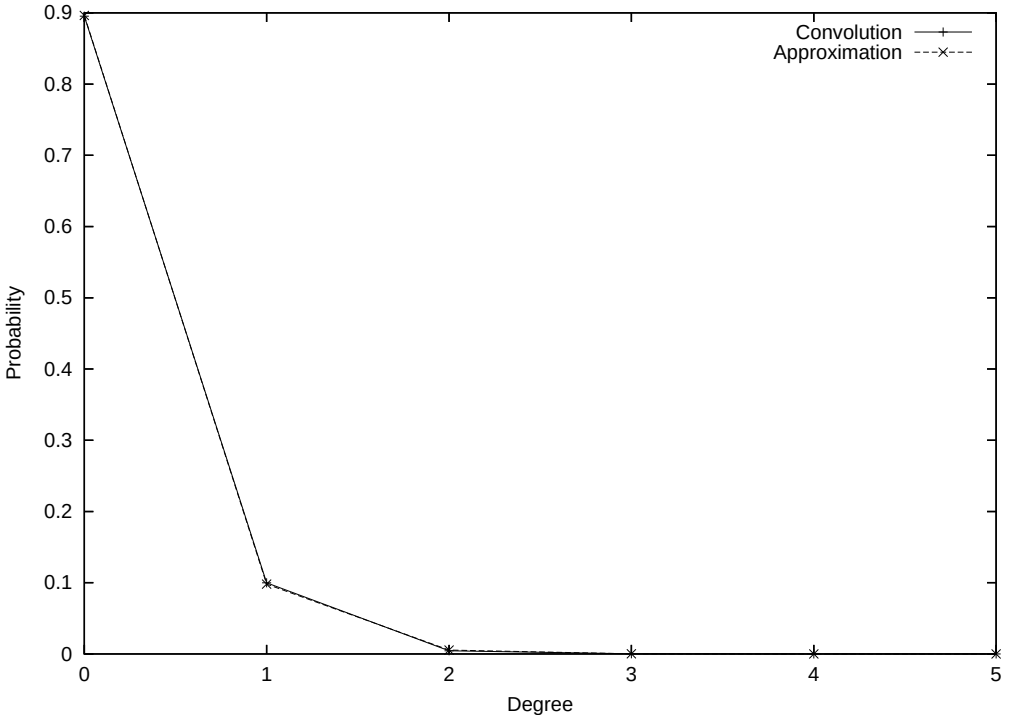
3.2 Optimizing graph generation

In this section we show how the underlying Kronecker structure can be used to achieve load-balance and cache efficiency in an RMAT-based Kronecker algorithm, and investigate both single-threaded and multi-threaded performance.

Firstly, note that the need to choose an edge count in RMAT a priori is obviated: the pmf for the total edge count of a Kronecker graph turns out to be extremely well approximated by a simple Poisson distribution. Many programming libraries



(a) The exact and approximate degree pmf for high-degree nodes.



(b) The exact and approximate degree pmf for low-degree nodes.

Figure 2: We show results of the experimental validation for two types of nodes: low-degree and high-degree, as defined in the text. The exact degree pmfs are computed by the convolution of binomials from theorem 3.5, and the approximations are the Poisson distributions from theorem 3.10.

support random number generation from various probability distributions; we have implemented our algorithms in C++ and use the Boost Random Number Library [20] to draw node degrees or total edge counts. Thus, we now view our RMAT algorithms as equivalent to the Kronecker algorithm, rather than approximations.

The original RMAT algorithm's single-processor performance suffers from an inherently irregular memory access pattern, because each subsequent edge is computed by a stochastic process and is independent of any previous edges. Furthermore, it is non-trivial to parallelize efficiently because each thread must frequently access other threads' regions of the graph and a naïve labeling scheme imposes significant load-imbalance.

In the discussion that follows, p denotes the number of threads in the computation.

3.2.1 Load-balancing

We can use the node categories to efficiently compute a load-balance for the graph generation algorithms. Recall that the time complexity of the RMAT procedure is $\mathcal{O}(|E|)$, so that each thread should be assigned a set of nodes with roughly the same expected number of total edges.

The load-balancing procedure is as follows. Conceptually, the categories are first sorted in descending order of expected degree. Then, using the expected number of edges a particular thread should handle ($\mathrm{E}[\mathcal{E}]/p$, eq. (4)), a simple calculation can add partial or complete categories to each node, going from left to right. Calculating the number of edges corresponds to computing the surface area in the expected degree plot. See figure 3 for a cartoon illustration.

Not only should each thread have roughly the same amount of work, for undirected edges the algorithm must also access other threads' regions to add the remote side of an edge. When the work load-balanced this overhead is minimized.

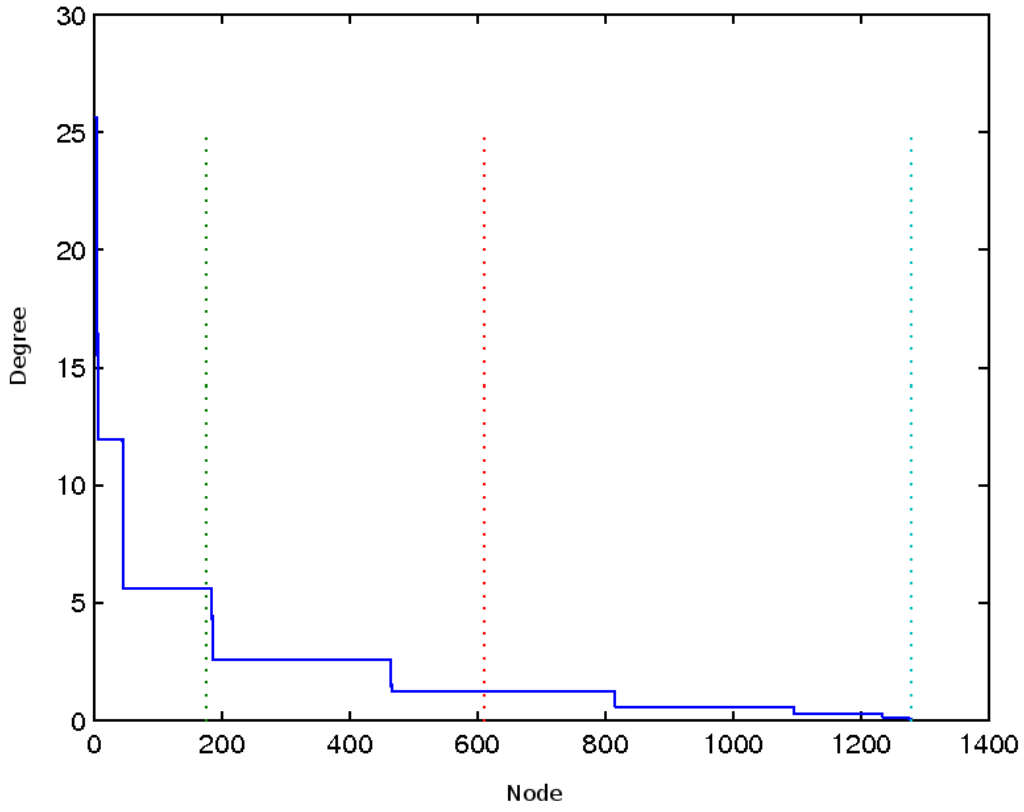


Figure 3: A cartoon illustration of achieving load-balance. All nodes are sorted on expected degree in descending order. By calculating surface areas of $\mathcal{O}(n)$ squares, each thread can be assigned an equal expected number of edges to handle. The solid line represents the expected degree of the nodes, and the dotted lines represent the boundaries of load-balanced work for three threads.

3.2.2 Memory and cache efficiency

For undirected graph generators, cache efficiency becomes even more important than for directed generators, which are much simpler. Cache efficiency for adding the initial sides of each edge is achieved as follows: nodes in a work region are accessed in sequential order, a node degree is drawn according to eq. (2) and its edge array is updated in sequential order. However, the remote edge sides must be added in another thread's region with probability $1 - 1/p$.[1] This contrasts with the directed case where no additional work is needed.

Because we assume a NUMA - where each processor may have its own memory module - accessing other thread's graph regions may have serious performance implications. For this reason all our RMAT implementations, except for the original one, queue an 'update' in one of p local caches whenever a write operation is needed in another thread's region. Threads thus operate in their local memory for most of the time and their graph data is never accessed by other threads. At intermediate steps, all local caches are copied to the appropriate shared queues using efficient 'batch' memory copy operations, and each thread can perform the received operations in local memory again. For larger graph sizes this technique could even enable distributed memory computation, but here we focus on shared-memory parallelization.

Without any extra work, the cached write updates are in arbitrary order and would impose an irregular memory access pattern once again. In our implementations we distinguish between performing those new updates immediately or first sorting all p shared queues in parallel.

3.2.3 Data structures and optimizations

A common way to store sparse graphs for computational efficiency is an array of adjacency arrays (see left side of figure 4). Edges of the same node that are accessed

[1] Assuming load-balance.

in order are cache efficient, but accessing edges of subsequent nodes most probably incurs a cache miss. Notice also that all adjacency arrays must be of the same size to accommodate efficient data access and prevent resizing.

Not only our graph algorithms, but also the set of epidemic dynamics algorithms process both the node structures[2] and their edges in order, so we attempt to optimize for this case and do better than the $\mathcal{O}(N)$ random accesses of the adjacency arrays.

We store each ith edge of a node x as the xth element in the ith separate array of edges. Thus, each ith array contains an edge for nodes that have at least i edges and an empty entry $\emptyset$ otherwise.[3] See the right-hand side of figure 4. As illustrated here, the renumbering of nodes makes it likely that an edge accessed for a node x is in cache, because the previous node $x-1$ has an expected degree that is equal or greater than that of x.[4]

In addition, the data structure can reduce the amount of storage that must be allocated to a minimum. Each array can be of different sizes: array k could cut off at $x_{\max}$ when the probability $d_{x_{max}}(k) \leq c$, where c is an suitably small probability.

3.2.4 Implementations

We describe four RMAT algorithms that we have implemented here, which are compared in sections 3.2.5.2 and 3.2.5.3.

3.2.4.1 Original RMAT

The original RMAT algorithm calculates and adds a new edge (x_i, y_i) $|E|$ times, where $|E|$ is a parameter of the algorithm. The initiator matrix Θ is normalized and for each edge one of four entries is chosen stochastically n times. When a new edge is nominated it is immediately added to the graph, i.e. two random accesses into

[2] In the node structure we store information such as infection status, node type, degree etc.

[3] Actually, node structures already contain the degree of a node so that all edge structures that are not used by a node need not specifically reset.

[4] In our simulations the required storage for an edge side is 6 bytes, so a few consecutive lower degrees than that of x have no impact on cache-efficiency.

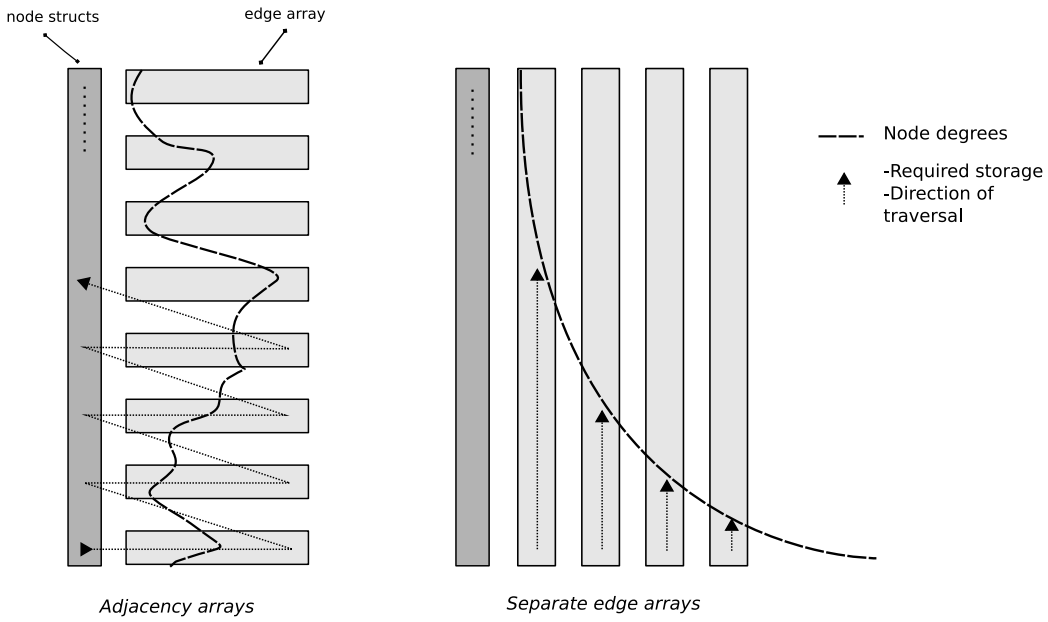


Figure 4: Two possible graph data structures. On the left: a node structure contains both node variables and an adjacency array, and all node structs are stored contiguously. Each node must allocate space for the maximum possible degree. On the right: edge structures are stored separately, in m separate arrays for maximum degree m. Nodes are sorted on descending expected degree, and edge traversal across nodes is sequential.

the graph structure are performed. Note that this algorithm most likely saturates memory-processor bandwidth as none of the edge calculations and updates depend on the previous edges. However, some bandwidth is wasted because data that is brought in for cache efficiency is almost never used.

When using multiple threads, this algorithm must use node-based mutexes to avoid inconsistencies.

3.2.4.2 Optimized RMAT variants

All the algorithms described here have the following stages in common:

1. Each thread processes its nodes sequentially, drawing a random number for its degree (according to eq. (2)). Local edge side updates are applied immediately, and remote updates are queued in one of p local queues.

2. All threads perform a batch copy from each ith local queue to the shared queue of thread i.

3. Each thread performs the received updates from its shared queue in its local region. Update failures (e.g. because a node is full) are cached in local queues as before, for the appropriate thread.

4. All threads perform a second batch copy from the local to the shared queues.

5. Each thread performs the corrections from its shared queue in its local region.

The variants and their names are:

CRMAT No load-balancing is used, and no sorting is performed.

LRMAT Load-balancing is used, but no sorting is performed.

SRMAT Load-balancing is used and sorting is performed.

Note that none of these need node-based mutexes because each thread is assigned a disjoint region of the graph.

3.2.5 Performance comparison

We assess the performance of the implementations in two sections. First we look at single-threaded performance to assess the cache efficiency from applying updates in order, and then we attempt to scale up to more threads. In all experiments, both RMAT and CRMAT did not use the renumbering scheme (nodes ordered on descending degree) because they do not load-balance and would strongly localize their memory accesses.

All implementations were done in C++, parallelized with `OpenMP` and compiled with `GCC 4.3` using optimization `O3`. The architecture consists of four AMD Opteron 2 GHz quad-cores, each with a 512 MB cache. The system operates under Linux 2.6.9.

Lastly, in all performance plots we zoom in on the largest graph sizes; for smaller graph sizes the algorithms perform comparably.

3.2.5.1 Impact of refactoring the graph data structure

In all experiments that we present, the refactored data structure (on the right of figure 4) is used because the adjacency arrays performed much worse for all experiments. In particular, it turns out that the original RMAT benefits greatly from the refactoring, even though it was not the original motivation. The single-threaded version becomes an order of magnitude faster; when using more threads, the adjacency arrays made performance worse than the single-threaded version whereas with the refactored data structure the speed-up is superlinear. We do not attempt to explain the superlinear speedup.

We believe that the single-thread speed-up of RMAT using the refactored data structure comes from the fact that high-degree nodes incur less memory-contention. Each consecutive edge of a node is stored with $\mathcal{O}(N)$ other nodes' edges in between, and for a high-degree node the chance that the same edge is accessed by two or more

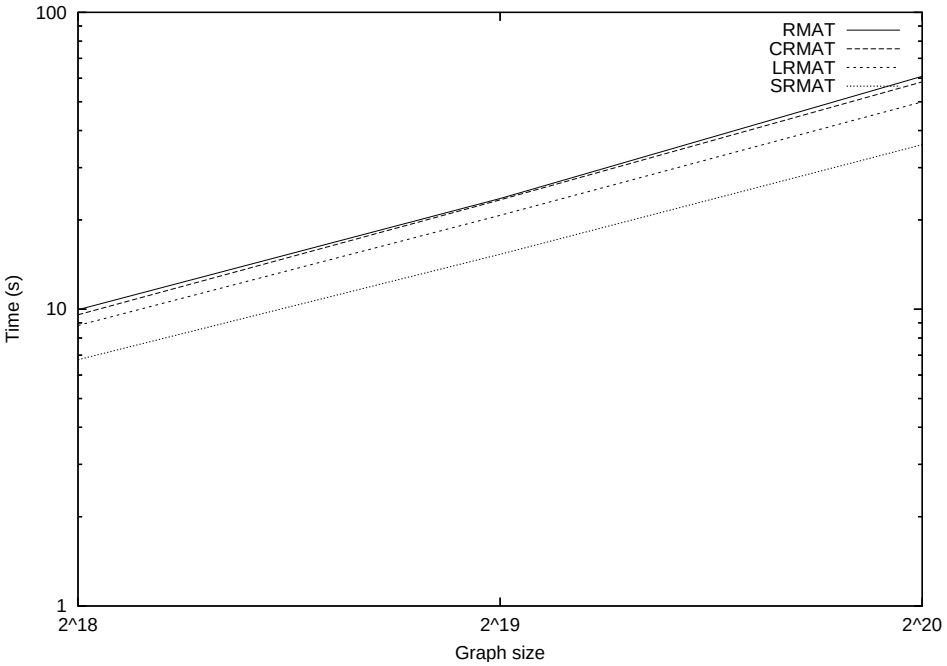


Figure 5: Performance comparison of the graph generators using 1 thread. The running time reduction is up to 220% and is due to renumbering and sequential application of updates. As the graph size doubles, the running time of RMAT scales by a constant of about 2.5 and that of SRMAT by about 2.3. This plot includes only graphs with sizes of powers of two.

threads is relatively small.

3.2.5.2 Single-threaded performance

The effects that we have investigated are: renumbering and sequential application of initial local updates, and sorting the shared queues after the batch copies. The timing results for larger graph sizes are in figure 5. Note that the performance gains discussed here are in addition to the order of magnitude discussed in the previous section.

Our efforts have paid off with a satisfactory speedup for SRMAT: 220% for the largest graph size that we have generated (2^{25} nodes). LRMAT outperforms RMAT by about 150%, i.e. merely renumbering the nodes and creating edges for subsequent nodes is already more efficient. In addition, SRMAT sorts the shared queues after the intermediate batch memory copies, accounting for the remaining performance

gain. Lastly, we observe that CRMAT performs comparably with RMAT which shows that the performance gain of the other two algorithms lies in the combination of renumbering and sorting; CRMAT performs the initial local updates in sequential order as well, but due to irregular degree patterns the cache efficiency is negligible. (See also figure 3.)

3.2.5.3 Multi-threaded performance

We have also run the algorithms with 4 and 16 threads for the same graph sizes; the results are in figure 6. We find the same relative performance of CRMAT and LRMAT as in single-threaded experiments, so we only show RMAT and SRMAT for simplicity.

Interestingly, the performance benefit of SRMAT is comparable to but less than that found in the single-threaded case: 185%. For the graph sizes and the two parallelization factors that we have used, RMAT has a superlinear speedup and SRMAT has an almost perfect speed-up. However, from figure 6 it appears that the trend of SRMAT's running time grows more slowly. We could not experiment with larger graph sizes and thus do not know if this trend continues beyond 2^{25} nodes.

3.3 Other applications

The novel understanding of the structure of Kronecker graphs can be used in many different contexts, of which we already exploit several in chapter 4. We plan to investigate these and other applications in more detail in the future: for now, what follows should be regarded as the author's belief.

3.3.1 Posterior graph probability

In a few applications the problem arises of computing the probability that a particular graph is generated by a given generator. Leskovec et al. [15] attempt to fit four

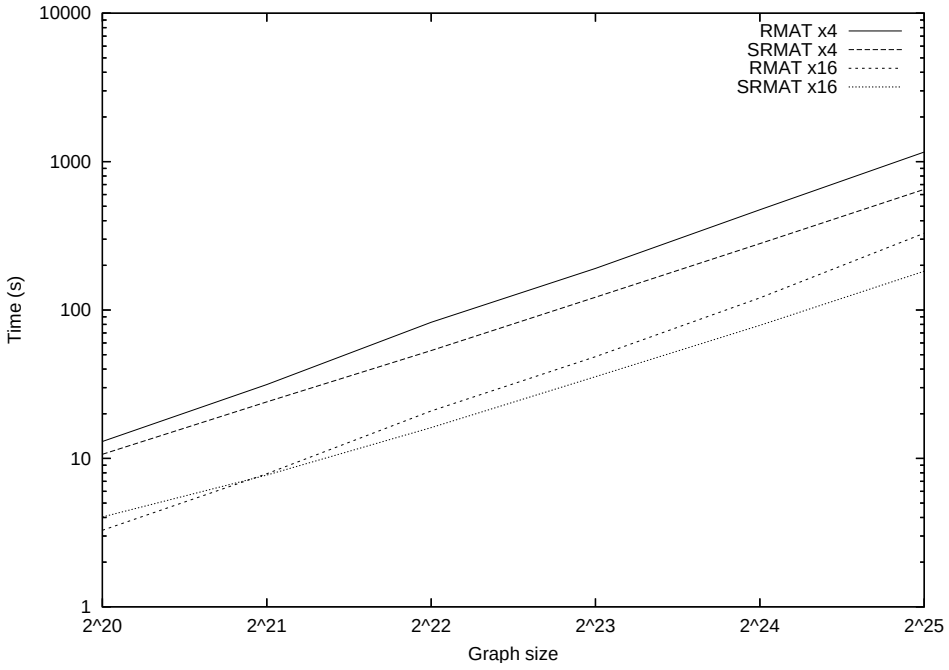


Figure 6: Multi-threaded performance comparison between RMAT and SRMAT. For both 4 and 16 threads, the difference in performance increases as the graph size is increased. For the largest graph size that we have experimented with, the performance gain is about 185%. As the graph size doubles, RMAT's running time scales by a constant of about 2.5 and SRMAT's running time by about 2.15. This plot includes only graphs with sizes of powers of two.

Kronecker parameters to an existing graph. He approximates an exact fitting algorithm of superexponential time complexity using sampling and gradient descent in parameter-space, and achieves $\mathcal{O}(|E|)$ performance (with an extremely high constant factor) for the purpose of parameter fitting.

Using the $n+1$ node categories, however, the problem of computing the posterior graph probability degenerates to that of a trivial *classification problem*. We expect that nodes need only be classified based on their degree because choosing the second side of an edge follows exactly the same probability distribution over the node labels regardless of the initiating node (corollary 3.3). Intuitively, an edge that connects to an 'unexpected' node label is accounted for in an 'unexpected' degree or a different classification for that node.

This already yields an $\mathcal{O}(N)$ algorithm per posterior calculation, but for parameter fitting we can improve this to constant amortized time complexity as follows. If the nodes of the target graph are sorted on degree once, they can be assigned to categories in contiguous ranges[5] and using eq. (2) all probabilities can be calculated.

Anecdotal experiments suggest that this approach is both accurate and extremely fast. For a network of autonomous systems consisting of $14,000$ nodes we evaluated all a, b, c, d parameters using exhaustive search with step size 0.001, and found parameters equivalent to [15] in less than a second on an ordinary laptop.

In this dissertation we are concerned with whether the network dynamics operators presented in chapter 4 maintain graph properties over time in the absence of a virus. For this purpose we use the heuristic to automatically classify nodes to the category for which they were generated for ease of implementation.

[5]Here we use eq. (1) which specifies the size of each category.

3.3.2 Supporting arbitrary graph sizes

A common issue with the Kronecker algorithm is that the graph size must be a power of 2. To our knowledge, no successful efforts to interpolate the graph size are available in the literature.

If N may be any arbitrary number, then n is real-valued. Whatever the interpolation method, at some point it must transition from $\lfloor n \rfloor$ to $\lceil n \rceil$ categories. We suggest to use $\lfloor n \rfloor$ categories if $n < \lfloor n \rfloor + 0.5$, and $\lceil n \rceil$ categories otherwise. The degree pmfs are already defined for real-valued n, and for the category sizes the generalized binomial coefficient could be used. Having generalized all relevant equations that define the Kronecker algorithm, arbitrary graph sizes could be generated.

3.3.3 Mathematical modeling on Kronecker graphs

The underlying structure turns out to be very compact, making it feasible for mathematical models to incorporate all graph properties that the Kronecker algorithm provides. A naïve way of modeling population network properties could fit a power-law distribution at most, and would result in a large array of equations. In contrast, distinguishing between a mere $\log_2 N$ sets of nodes, each characterized by two properties (eq. (1) and (2)), a mathematical model can make use of all the properties of Kronecker graphs, in particular power-law distribution, community-structure and a small-diameter.

We use this technique in section 4.5 where we validate our framework against a mathematical model for a simple simulation.

Chapter IV

SEECN: A FRAMEWORK FOR SIMULATING EPIDEMICS

We maintain that for accurate modeling of epidemics through population networks, the dynamics of both the epidemic and the population must be taken into account simultaneously. In this chapter we discuss the processes of social network dynamics, virus propagation and virus progression. Although the framework extends to other applications, we focus on infectious diseases in this dissertation.

Combining the three processes into a single model has rarely been done before. Eubank et al. [13] constructed a very accurate population network of an urban environment and explicitly modeled social dynamics, and Moore and Newman [22] discusses the impact of the small-world property on the percolation of epidemics in general. More recently, upon the rapid and seemingly random spread of SARS, Meyers et al. [21] explicitly argue that social network features and dynamics must be considered for adequate modeling and that previous models failed to explain the spread of SARS for this reason.

In this chapter we adopt the convention of Sloot and Ivanov [25] and define stochastic 'operators' that take a graph instance as input, manipulate it and output the resulting graph. In particular, we compute the dynamics of infectious diseases by applying a composite network operator $G(t+1) = \Gamma G(t)$ to the graph $G(t)$ of a time step t, where a time step is the equivalent of some arbitrary duration. We use $G(t) \equiv \langle V(t), E(t) \rangle$ for ease of presentation. Γ is equivalent to the ordered application of distinct operators $\prod_i \Gamma_i$, each with a different function. In this way we isolate all dynamics from each other, facilitating model specification and incremental

development.

We motivate and define the network dynamics and operators in section 4.1, and the epidemic dynamics and operators in section 4.2. Then, in section 4.3, three different implementations are discussed which are compared for performance in section 4.4. Finally, we define and simulate a simple model in section 4.5 and validate the simulator by comparing the incidence statistics against that of a mathematical model.

4.1 Social network dynamics

4.1.1 Stability

In modeling social network dynamics we assume that the graph generator is already parameterized to generate graphs that are faithful to the real-world graph. We further assume that, in the absence of a virus, the graph would maintain its global properties over time even though the exact structure may change significantly. More precisely, we assume that the graph should maintain roughly the same posterior probability of being generated by the Kronecker algorithm with the same parameters.

A social network may lose its structure through various influences - in our case epidemics. However, when such external influences are discontinued a graph should regain its original structure steadily over time. Again we demand that the posterior probability of being generated by the Kronecker algorithm should converge back to its original level. We refer to a model that satisfies all the aforementioned conditions as being 'stable'.

Lastly, we assume that the total population size does not change in all simulations presented in this dissertation. However, dynamic population sizes could be approximated by adding a special 'death' node status for which all interaction parameters are set to zero.

Both the simple model of section 4.5.1 and the more complex models of chapter

5 have been observed to be stable,[1] with and without the epidemic, but we omit the figures. The reason that the epidemic does not change the graph structure is that we have assumed equal connectivity for all node types and statuses, because SEECN's current implementation does not support dynamic assortativity. This seems to have significant impact only on nodes with AIDS, for which the transmission probabilities are simply set to zero.

4.1.2 Network operators

Important processes in social networks are the movement of individuals and changing 'social interaction' - whichever interaction may be relevant for the spread of the specific epidemic. In this dissertation we take HIV as our prototype, and the most prevalent means of propagation is sexual interaction in the homosexual community. However, the following generalizes to other epidemics as well.

In SEECN we distinguish between a node's *type* and a node's *status*. A type $t(x)$ of node x is a parameter that is assumed not to change or change infrequently, such as a node's gender, genotype, sexual nature etc. The status $s(x,t)$ is time-variant and could assume values such as healthy, infected, and isolated. These two variables are used to index into predefined, but possibly time-variant parameter matrices that define the various dynamics.

In the current implementation we model a population's temporal dynamics by adding and removing nodes and edges with some probability, depending on the involved nodes' variables (type and status). This is achieved with the following two operators:

Node replacement ($\Gamma_{\mathbf{node}}$) A Kronecker graph must maintain its exact size and we enforce this by immediately replacing a node that is removed from the graph. An infection status of 'isolation' or 'death' typically means that the node no

[1] We used the posterior calculation procedure described in 3.3.1.

longer participates in social interaction, but is not necessarily removed from the network instantly.

Nodes are removed from the graph in each time step with some probability indexed by the node's variables: $P\left(x \notin \Gamma_{\text{node}} G(t) \mid s(x,t), t(x)\right)$. For instance, nodes with status AIDS should be removed with much higher probability than healthy nodes.

It is important to distinguish between node replacement and node movement. In real population networks, n node may move and transport a virus to other locations. Our current implementation does not support movement explicitly; we assume that such non-local interactions are adequately modeled by the Kronecker's hierarchical community structure.

The new node is added to the graph as if it were generated in the generation phase and had not been influenced in all previous time steps. The node variables are drawn with prespecified prior probabilities.

Edge deletion/addition (Γ_{edge}) Edges represent the social interactions that are relevant to the epidemic and typically change over time. We do not simultaneously replace edges that were removed, which allows for nodes to have changing degrees over time. For stability, the addition and removal probabilities should balance each other in absence of the epidemic.

The probability that a particular node removes an edge is indexed by the type and status of both sides of the edge: $P\left((x,y) \notin G(t+1) \mid t(x), s(x,t); t(y), s(y,t)\right)$. However, the Kronecker algorithm has no straightforward and efficient way to support dynamic assortativity in choosing edge neighbors. Therefore the probability that an edge is added is only indexed by the local node variables, and its remote node label is chosen as in the graph generation phase (lemma 3.1), thus $P\left((x,y) \in G(t) \mid t(x), s(x,t); a, b, c, d\right)$.

More precisely, a new 'node degree' d' is drawn from a node's degree pmf and $\text{Bin}\,[d', p_{\text{add}}(s(x), t(x))]$ edges are added to the graph, incident on the node. The intuition is that high-degree nodes should add more new edges per time step than low-degree nodes, and the relative difference is taken from the Kronecker parameters because it is assumed that these are realistic.

4.2 Epidemic dynamics

An epidemic evolves in two ways: nodes may infect other nodes (external propagation) and an individual node's infection status may progress (local progression). Note that the parameters for these dynamics are specified independently from those of the social network dynamics, but they both interact. An epidemic can only spread from one node to another if an edge is present, and the infection probability as well as the local progression depend on the node variables. In the opposite direction, an infection status influences a node's connectivity and 'life expectancy'. These interactions are not specified explicitly.

Local progression is a special case of dynamics and are not adequately modeled by (memoryless) geometric distributions. For instance, people with HIV take an expected thirteen years to progress to AIDS, and rarely less than six years. Further, depending on a person's treatment and behavior the expected progression time can become twenty-two years.

We accommodate this by introducing a *progression time* $p(x)$ of the node with label x, which is the minimum number of time steps before a node is considered for progression again. In our current implementation we can combine a normal distribution for the progression time and a geometric distribution afterwards, and obtain a difference between the median and mean AIDS incubation time [18]. Note that a node may still receive external infections or be replaced.

External propagation (Γ_{infect}) Nodes may infect their neighbors in a time step,

with probability indexed by both node's type and status. For a fixed x and all edges (x, y_i), $P\left(s(y_i, t+1) \mid t(x), s(x,t); t(y_i), s(y_i,t)\right)$

Local progression (Γ_{progress}) A virus can progress within a node itself without any external influences, and follows $P\left(s(x, t+1) \mid t(x), s(x,t), p(x)\right)$. A new $p(x)$ is determined after every change of infection status.

4.3 Implementation of operators

For our implementation of all network operators we use similar techniques as in the graph generation phase: data structure refactoring, load-balancing, caching, batch copies between threads and sorting. We have implemented three different versions: cached, load-balanced and sorted (which implies load-balance). We compare them for performance because we expect to use the simulator for parameter-searching and hill-climbing techniques in the near future.

A direct implementation (i.e., no caching) was omitted because it is not possible to offer the same semantics as the cached versions. More specifically, nodes that are not yet processed may become infected *and* infect other nodes in the same time step. This means that nodes that are processed at a later time have a much higher probability of infecting others.

The order of implementation does not matter over multiple time steps. We have chosen $\Gamma \equiv \Gamma_{\text{infect}}\Gamma_{\text{progress}}\Gamma_{\text{node}}\Gamma_{\text{edge}}$. Ordering epidemic operators before network operators have the advantage that the network operators may use the new variable values immediately, instead of performing updates twice. Also note that replaced nodes already have their degree recomputed and are not considered for edge dynamics.[2] All implementations share the following logical stages:

1. Nodes infect their neighbors with some probability, and queue an update for the neighbor's new variable values if they change.

[2]The edge operator operates over $G(t)$ and a replaced node is strictly a member of $G(t+1)$.

2. Nodes undergo local progression of their status variables, if applicable.

3. Nodes undergo the network dynamics operators. First it is possibly deleted and replaced by a fresh node; otherwise each of its existing edges is considered for removal. Then, a new 'degree' d' is drawn from the node's degree pmf and considers adding an edge d' times.

4. All local caches are batch copied to the appropriate shared queues. These caches contain e.g. updates for edge sides and new node variables.

5. The variables of all nodes are changed according to the updates they receive. In general, multiple but distinct variable updates may be received so we use a status conflict resolution procedure that is specific to the epidemic that is modeled. We expect that in most cases, a total order of precedence on all possible statuses can be imposed; for instance, 'infected' precedes 'healthy', and 'death' precedes 'infected'. In other cases, superpositions may be defined such as 'infected and treated'. For each node that changes node variables, new updates must be cached for all remaining and new neighbors.

6. All edge configurations are changed as a result of the network operators, and new updates are cached when e.g. an edge addition fails.

7. All local caches are batch copied to the appropriate shared queues a second time.

8. The corrections from the shared queues are applied.

They differ as follows:

Cached Merely cache updates locally and batch copy them to the appropriate shared queues at intermediate stages. Nodes are not renumbered.

Balanced Nodes are renumbered and accessed in descending order of expected degree. Workload is balanced over all threads; because performance is dominated by the $\mathcal{O}(|E|)$ updating procedures, this corresponds to the same technique from section 3.2.

Sorted The same as 'balanced', but upon completion of the intermediate batch copies all shared queues are sorted on node id. Thus, even updates in later stages are applied in sequential order as much as possible.

4.4 Performance comparison

We show the performance gains of the balanced and sorted algorithms over the cached in figure 7. Using 1 thread, the performance gain for the balanced algorithm is 155% and for the sorted algorithm 210%. When using 16 threads, the performance gains become 175% and 250%, respectively. Parallelizing the cached algorithm with 16 threads reduces the running time by a factor of about 2.1; for the sorted algorithm this factor is about 3. Lastly, the running time appears roughly constant over time, which is due to the stability of the operators as discussed in section 4.1.1.

The better scalability of the sorted algorithm is most likely due to the fact that all threads are operating in their local memory, except for the intermediate batch copies. The scalability of the balanced and sorted algorithms has been observed to improve as the graph size increases, but we could not experiment with graphs of more than 2^{20} nodes with a single thread.

4.5 Experimental validation

To convince ourselves of the correctness of the simulator, we performed a simple simulation that can also be modeled adequately with a mathematical model. This model sums out all possible infection sources but does account for Kronecker graph properties. We compare the plots of the infection incidence per time step.

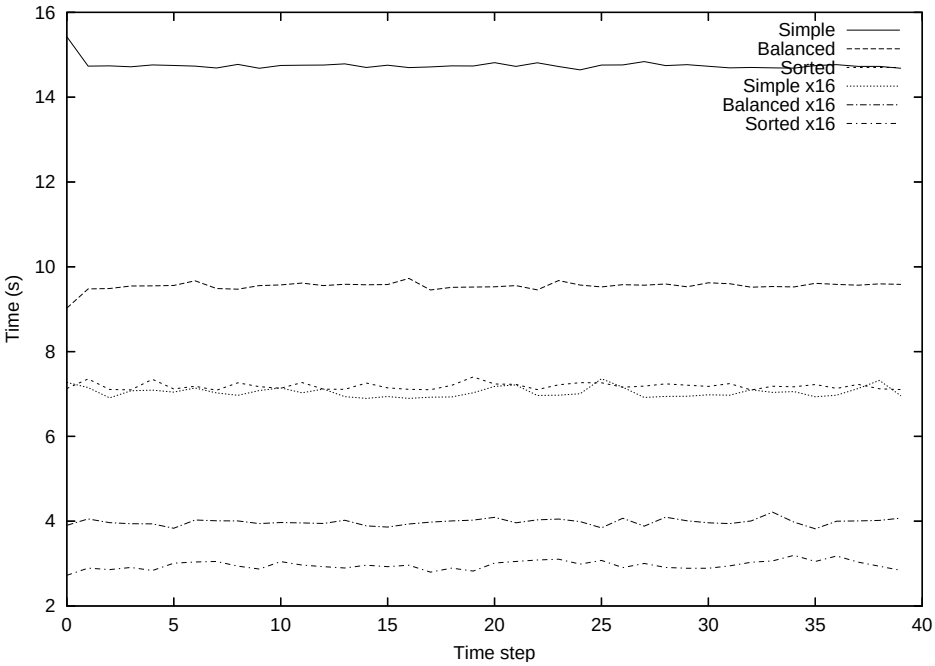


Figure 7: The running time of the three implementations of the operators, for 1 and 16 threads, on a graph of 2^{20} nodes. This figure also shows that the running time of simulating an epidemic remains constant over time.

4.5.1 Simple simulation setup

In the simple simulation there is only one infection probability $\iota = 0.15$ and each node has the same type. The infection status is either 'healthy' or 'infected'.

Node replacement is done with probability 0.05 (i.e. they exist for an expected 20 time steps), and edges are deleted and added with probabilities 1.0, to approximate non-locality[3]. This is because the mathematical model cannot express epidemic locality other than across node categories. To contrast, we also include an incidence plot where edges are not replaced at all.

4.5.2 Mathematical model

Because the purpose of the model is to validate the generator, we specify it to incorporate a Kronecker graph structure. Again we can use the results from section 3.1

[3]We define "epidemic locality" to be the rate of change in the set of susceptible (healthy) nodes that have edges towards infected nodes.

and model the progression of infection incidence for $n+1$ categories of nodes, making the model tractable and still accounting for all graph properties.

Let $\Omega_i(\mathcal{C}_b)$ be $\omega_j(i)$ of any node i in category $\mathcal{C}_b$. If $N = 2^n$ is the total population count, then $|\mathcal{C}_b| = \binom{n}{\Omega_1(b)}$ is the number of nodes in category b. Also, let $f_i \in \{a, b, c, d\}$, $T = \sum_i f_i$ and each $\tilde{f}_i = f_i/T$.[4]

We further define $\rho(t)$ as the total infection prevalence in time step t and $\rho_b(t)$ that of category $\mathcal{C}_b$, $\iota = 0.15$ as the independent infection probability and

$$\mathrm{E}[|E_{o\leftrightarrow p}|] = \binom{n}{o} \min\left[(a+b)^{n-o}(c+d)^o, k_{\max}\right] \cdot \binom{n}{p} (\tilde{a}+\tilde{b})^{n-p}(\tilde{c}+\tilde{d})^p,$$

where $k_{\max} = 120$ is the maximum degree of a node. The model then becomes:

$$\Delta\rho_b(t) = \sum_{o=0}^{n} \left(|\mathcal{C}_b| - \rho_b(t)\right) \times \left(1 - (1-\iota)^{\frac{\mathrm{E}[|E_{b\leftrightarrow o}|]}{|\mathcal{C}_b|} \frac{\rho_o(t)}{N_o}}\right),$$

and

$$\Delta\rho(t) = \sum_{b=0}^{n} \Delta\rho_b(t).$$

4.5.3 Fitting result

We plot $\rho(t)$ and the results of the two simulations (0% and 100% edge dynamics) in figure 8.

The incidence plots of the mathematical model and the simulator with 100% edge dynamics are virtually indistinguishable, so the simulator appears to be valid. The only operator that we could not validate is Γ_{progress}, because the mathematical model is memoryless for all node statuses. We did not find any obvious way to extend the model with progression. However, in chapter 5 we find encouraging results that fit reported statistical data with remarkable quality.

The effect of epidemic locality is also clearly visible. The model has virtually no locality because it does not specify edge configuration beyond that of category

[4] In other words, $\tilde{a}, \tilde{b}, \tilde{c}, \tilde{d}$ are the normalized RMAT parameters.

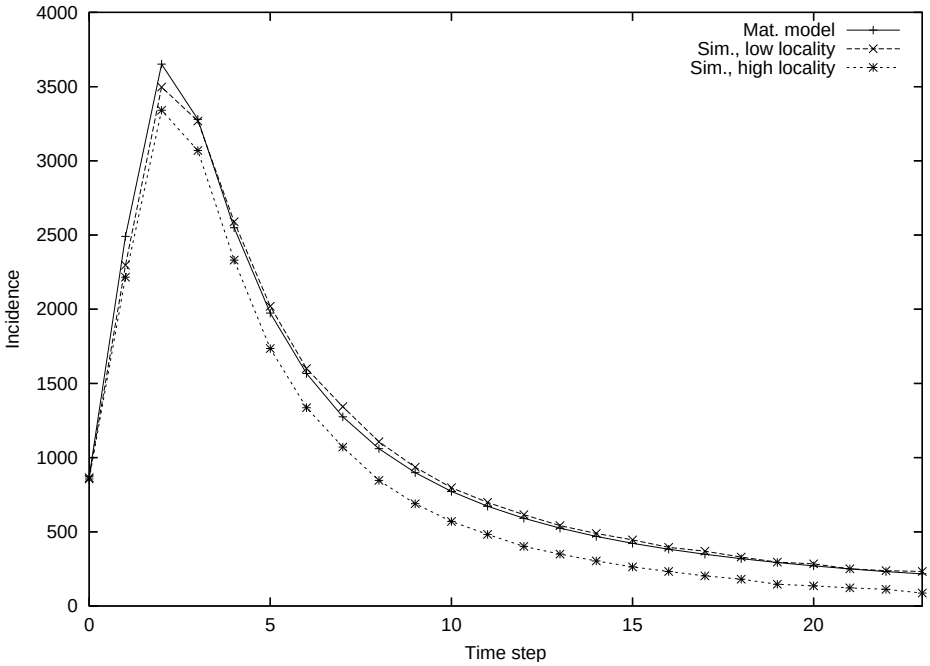


Figure 8: The incidence plots of the mathematical model and two simulations, one with 100% edge dynamics and the other with 0% edge dynamics. The former is almost indistinguishable from the model, and the latter shows the effect of 'epidemic locality'.

combinations, and is approximated over time with 100% edge dynamics. The second simulation (0% edge dynamics), however, has maximum epidemic locality (modulo the replacement of nodes). The effect is a slower epidemic evolution.

Chapter V

SIMULATIONS

We use the SEECN framework to perform two relatively realistic simulations of the HIV epidemic in the homosexual community. Different aspects of HIV have been modeled by numerous researchers, almost exclusively using Markov models and differential equations. To our knowledge, Sloot and Ivanov [25] are the first to simulate the HIV epidemic using complex networks and explicitly modeled the sexual interaction network with a power-law distribution. However, their graphs are random in every other respect and regenerated at every time step, resulting in virtually no epidemic locality. Also, local progression of nodes is only modeled using a geometric distribution.

In this chapter we model the effects of different infection stages, diagnosis, treatment and expected duration of sexual relationships simultaneously. We take or derive all parameters from the literature; they are specified in appendix A and summarized in section 5.1. Then we summarize observed trends in the incidence and prevalence of HIV and AIDS over time in 5.2; this offers a frame of reference for interpreting the results in the subsequent sections.

The first simulation (section 5.4) does not incorporate treatment and shows the basic evolution of the epidemic. The second simulation adds the notion of treatment and its results are discussed in section 5.5; in particular, we discuss differences with the results from section 5.4 and find remarkable resemblance of the epidemic evolution with available real data. Moreover, we highlight a some examples of conclusions in previous work that we consider invalidated based on our results.

5.1 Our model for HIV

The model that we use in the subsequent simulations is by no means meant to fit reported statistics quantitatively, however we do derive all our parameters from the literature. Our intention is to show the qualitative evolution of a realistic simulation and assess the relative impact of treatment. A detailed listing of the parameters of the model and their sources or derivations can be found in appendices A and B.

5.1.1 Epidemic parameters

Our domain is the homosexual community, so each node has the same type (male). The infection status can be one of healthy, acute HIV, untreated HIV, treated HIV and AIDS. We assume that an untreated HIV patient has an expected progression time into AIDS of thirteen years, and a slightly lower median. The expected progression time of a treated HIV patient into AIDS is taken to be about twenty-two years. Acute HIV lasts for approximately three months, which we take as the length of one time step.

Treatment uptake is typically modeled as a percentage of untreated patients per quarter. We calculate the uptake to ensure that about 70% of a node's time of infection is under treatment, and assume that 30% of the treatments fail. In the first simulation the uptake is set to zero. Various drug treatments have been introduced at different times, but in our simulations we model only one treatment type that is available at time step zero, and the use of which gradually increases.

We further assume that about 60% of untreated infected nodes is actually diagnosed and aware of their status, which results in 25% less risky behavior. In the second simulation, the availability of treatment results in a overall 25% increase of risky behavior and reduces infectivity of successfully treated nodes by 75%. Nodes in the acute HIV stage are much more infectious due to a peak of viral body count.

5.1.2 Network parameters

We use the Kronecker initiator matrix

$$\Theta = \begin{pmatrix} 0.945 & 0.632 \\ 0.632 & 0.051 \end{pmatrix},$$

and defer details of the calculation to appendix B. These parameters are estimates for a typical homosexual community for the purpose of a qualitative study of epidemics.

5.2 Summary of trends in reported data on HIV and AIDS

5.2.1 Purpose and context

Here we discuss some prevalent trends so that the reader is better able to interpret the plots of epidemic evolution in the subsequent sections. It is not meant to be exact, nor are we interested in explaining or correlating the observed trends here. Exact statistics on the incidence and prevalence of HIV and AIDS differ across cities and countries, however roughly the same trends were observed in industrialized countries. In this section we focus on the statistics in the homosexual community.

Please note that all reported data should be considered as approximate, especially that of HIV. The moment of infection of a person with HIV is unnoticeable, and most of the incubation period is symptomless. The rates and methods of diagnosis vary over time and it is difficult to assess how long ago seroconversion took place. The data for AIDS are more reliable, and we will use an actual plot of AIDS incidence to compare our results in section 5.5.

5.2.2 Trends in incidence and prevalence

A strong peak of HIV annual incidence was observed around 1985, which stabilized around 1990 [8, 24]. From 1990 the annual incidence has been roughly constant for a few years after which it started increasing by a small amount each year. Data from before 1985 are usually not available or very unreliable.

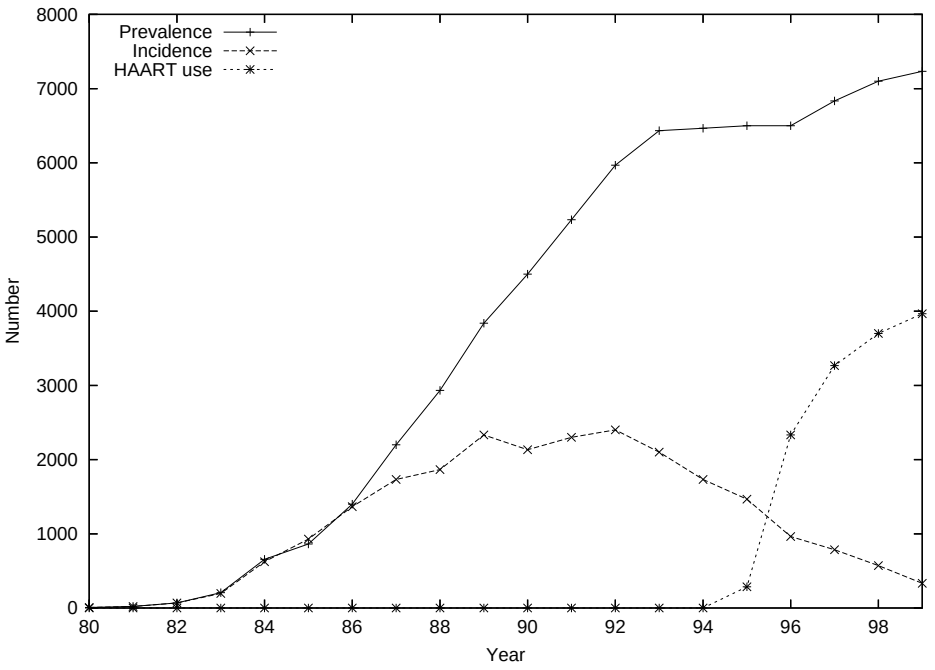


Figure 9: Trends of AIDS incidence and prevalence in San Francisco, and the use of HAART. Source: [14].

The incidence of AIDS started to rise gradually from about 1983 and plateaued for one or two years before 1995. Then it started to decline at different rates in different countries. The prevalence of AIDS plateaued for an extended period, and in some cases a small but distinctive second incline is observed. In figure 9, example incidence and prevalence of AIDS are shown for San Francisco.

5.3 Previous assumptions and conclusions

We highlight two papers in particular that have attempted to explain trends in the HIV/AIDS incidence and prevalence curves. Because of the higher confidence in reported data of AIDS, we will focus on those when comparing our simulation results. In particular, in sections 5.4 and 5.5 we compare and contrast our simulation results with the assumptions and conclusions that follow.

In [14], the countervailing effects of a specific effective treatment called HAART[1]

[1]Highly Active Antiretroviral Treatment.

are discussed. HAART reduces an individual's infectivity but also significantly increases the incubation time for AIDS, leaving more opportunity to spread the infection. In addition, they report that risky behavior among homosexuals has increased due to less concern about becoming infected because of the availability of better treatments. This work does not include any simulations but suggests that the positive effects of HAART appear to be counterbalanced or overwhelmed by the increase of unsafe sexual behavior. A second suggestion is that the second increase in AIDS prevalence (after 1996) is due to the introduction of HAART.

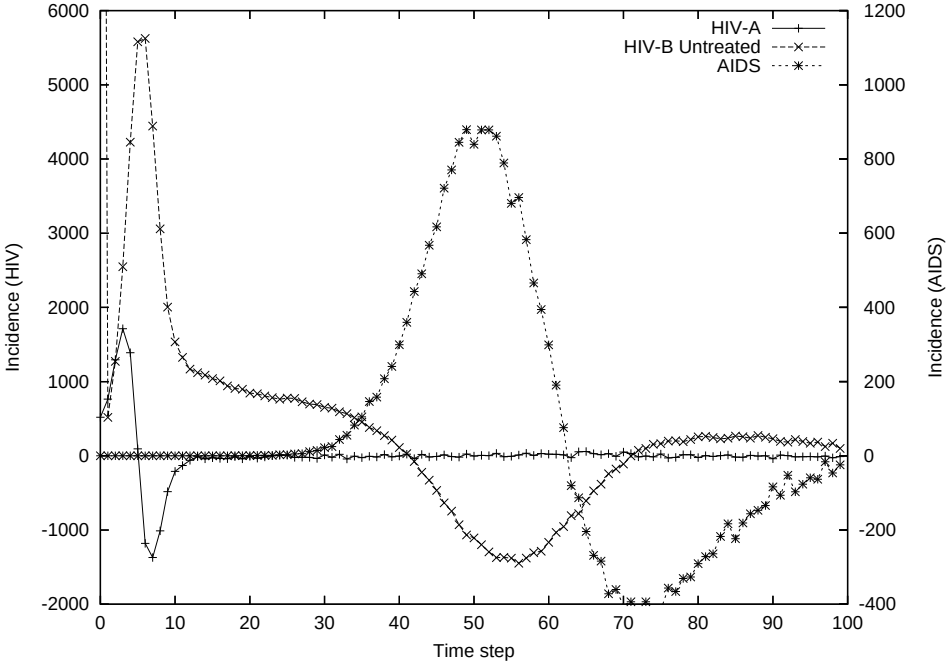
Aalen et al. [2] conclude that the decline in AIDS incidence is best explained by the introduction of new anti-retroviral treatment. For this conclusion they dismiss the influence of ordinary dynamics such as varying infectivity and depletion of susceptible people, based on the observations that the trends are comparable in different countries and the precipitousness of the decline. They use Markov models with different parameters to predict AIDS incidence.

5.4 Simulation without treatment

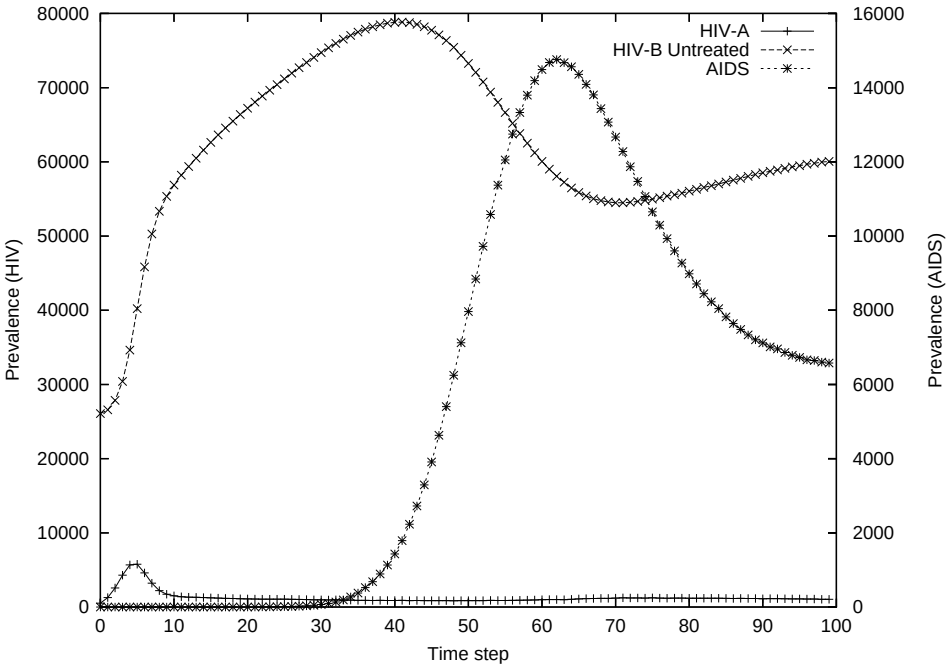
This simulation serves to show the basic shape of the incidence and prevalence plots for HIV and AIDS if no treatment were available. This enables us to better contrast the results of the next section. The resulting incidence and prevalence plots are shown in figure 10.

5.4.1 Observations

Interestingly, the simulation predicts a strong peak of HIV incidence in the initial phase, and a peak of AIDS incidence approximately eleven years later. Recall that we had set the median AIDS progression time at thirteen years and the expected progression time slightly higher. Also, the simulation predicts a short period of negative HIV incidence due to the progression to AIDS of a large number of nodes. If we increase the probability of node replacement this negative mode shifts to the right



(a) Incidence of HIV and AIDS.



(b) Prevalence of HIV and AIDS.

Figure 10: Incidence and prevalence statistics per time step of the simulation without treatment, averaged over four runs. Each time step corresponds to three months.

and becomes less prevalent, but it is still present. It is important to note that a negative incidence does not mean that no new nodes are infected with HIV; it means that more HIV-infected nodes are deleted than healthy nodes are infected.

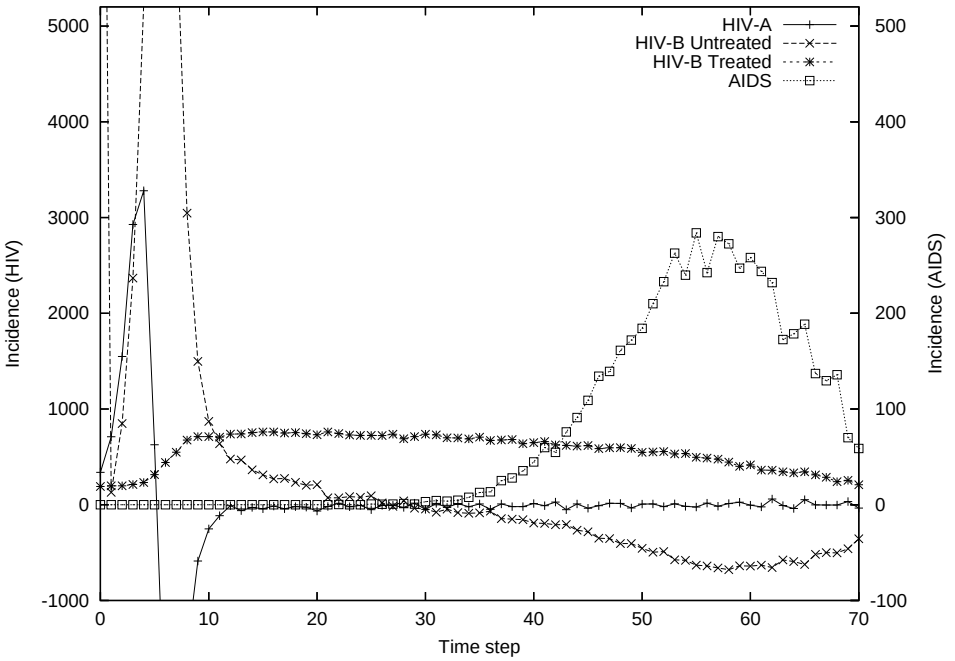
For the prevalence curves we observe a gradual peak for that of HIV, and a stronger peak for that of AIDS. After HIV's peak, a minimum is reached for a short period of time after which the prevalence grows gradually, and plateaus to some level. This growth is due to our assumption of constant population size and occurs when AIDS patients are deleted and replaced by new nodes. Because of our lack of knowledge of the homosexual population dynamics, we assume this is an artifact of the simulator's strict node replacement.

5.4.2 Conclusions

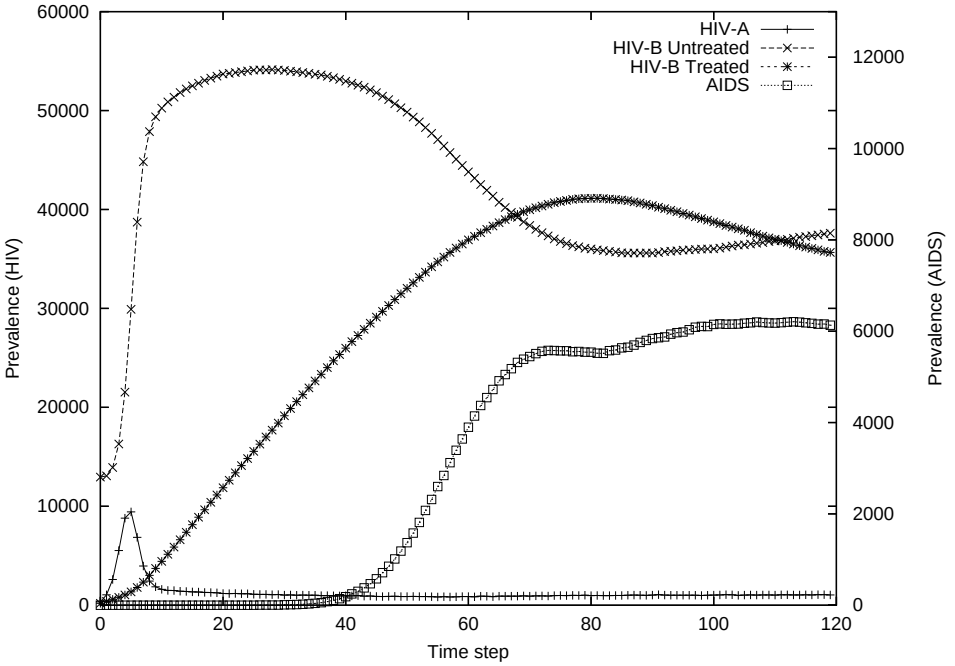
Although the simulation has no quantitative value we conclude that mere HIV epidemic dynamics among homosexuals, without the influence of any treatment or change in behavior, incur strong peaks in incidence data and more gradual peaks in prevalence data. Even though other influences may be present, we believe this underlying effect should not be dismissed in simulations that are used for explaining or predicting trends in the HIV epidemic.

5.5 Simulation with treatment

When we augment the basic HIV model with a constant quarterly treatment uptake the results already appear a good qualitative fit to reported data. The plots of incidence and prevalence are shown in figure 11. We have focused on the time steps of highest entropy and as far as seems representative of the reported data; note in particular that the incidence data is plotted over a different range for better detail. Recall that our model parameters were taken or derived directly from existing literature and were not fitted to reported data.



(a) Incidence of HIV and AIDS.



(b) Prevalence of HIV and AIDS.

Figure 11: Incidence and prevalence statistics per time step of the simulation with treatment, averaged over four runs. Each time step corresponds to three months. The data for treated HIV patients are overestimated by 30% due to treatment failure.

5.5.1 Observations

We observe that the extended mode in the incidence data for AIDS is reproduced by our simulation, as well as the following decline being equally steep as the incline before the plateau. The incidence mode is at the same position relative to the prevalence curve. The prevalence of AIDS rises over a period of approximately 8 years in our simulation, compared to 10 years in the reported data of figure 9, and both prevalence curves stabilize abruptly after the incline.

Looking more closely, we also observe that both curves share a slight but distinctive increase after the curves had stabilized for two and three years, respectively.[2] Katz et al. [14] correlate this with a simultaneous increase in the use of HAART.

In the HIV figures we find that the differences in model parameters for the treated simulation - decreased infectivity of treated nodes, increased risky behavior and increased progression time - result in a higher stabilized combined prevalence of HIV: approximately 125% of the results in section 5.4. Secondly, the HIV prevalence has a significant and gradual decline which ends around the time of the small but distinctive increase in AIDS prevalence - a striking resemblance to the reported data in [24] combined with that of figure 9.

5.5.2 Conclusions

Although the model is assumed to be only of qualitative value and is not designed to provide a complete model for the HIV epidemic, it is remarkable that the same trends - and deviations thereof - are predicted with high resemblance. The model augmented with treatment appears to be a good qualitative fit to the reported data.

Our results confirm the conclusion of section 5.4 that the mere dynamics of the HIV epidemic over a representative population network cannot be ignored in explaining and predicting trends in incidence and prevalence. In particular, Aalen et al.

[2]The increase remains after averaging over multiple simulation runs.

[2] assumed that the dynamics of HIV and the depletion of susceptible nodes could not cause a steep decline in AIDS incidence, and therefore attributed it to increased effectiveness of treatments. Regardless of what effect treatment may have on AIDS statistics, we consider their assumptions unfounded. To prevent mistakes we argue for simultaneous and realistic simulation of both the HIV epidemic, the population network and the introduction times and effects of treatments before adequate predictions of the evolution of HIV and AIDS can be made.

Further, we have reproduced the small but distinctive increase in AIDS prevalence after 1996 without the introduction of any treatment around that time. Katz et al. [14] suggests that the increase is due to the introduction of HAART; however, we find that this cannot be concluded until more realistic models are developed. Lastly, the paradoxical result that the introduction of treatment is eventually counter effective is in accordance with the suggestion of [14]. No observations can be made from cumulative statistics or periods of high entropy because of the different introduction times of different treatments, nor has the observed ratio any absolute significance.

Chapter VI

DISCUSSION AND CONCLUSION

We believe that traditional mathematical models are sufficient for relatively simple simulations but become increasingly intractable as more influences and interactions must be considered. Realistic simulations must incorporate the distinctive static properties of populations and the temporal dynamics of social interactions and epidemics simultaneously. In this dissertation we enable such simulations and argue to represent populations and their social interactions explicitly using annotated complex networks, and to factor different types of dynamics in separate operators that are applied to the network in every time step. Nodes are characterized by a static 'type', such as male, and a dynamic 'status', such as being infected and treated. All dynamics operators are parameterized independently of each other in terms of the involved nodes' type and status, which facilitates the development of complex models and the analysis of the relative impact of particular effects.

We present SEECN, a framework that combines efficient algorithms for graph generation and various operators for modeling temporal dynamics. In the current implementation, four types of dynamics are supported: replacement of nodes, addition and deletion of edges, internode epidemic propagation and intranode infection status progression. For generating graphs we adopt the Kronecker graph generation algorithm because it can be fit to real social networks with remarkable resemblance. Our implementation is based on the more efficient RMAT approximation, which we improve to be statistically indistinguishable. This enables theoretical results for either algorithm be used interchangeably.

The primary drawback of modeling epidemics in so much detail is a tremendous

increase in running time. For this reason we have spent considerable effort in speeding up graph generation and epidemics simulation, and find satisfactory results. Compared to a naïve implementation of an RMAT graph generator, our single-threaded implementation is an order of magnitude faster. The multi-threaded implementation scales almost perfectly, whereas the naïve RMAT does not benefit from parallelization. For a machine with 16 processors, the result is a speed-up of about two orders of magnitude. Running times for the dynamics operators cannot be compared to existing implementations, but their implementations use the same techniques as our improved RMAT algorithm. All in all, SEECN is practical: simulating a model of medium complexity over a graph of a million nodes for 30 time steps takes about ninety seconds.

As a prototype, we simulate two relatively complex models for the HIV epidemic and find a remarkable qualitative fit to reported data for AIDS incidence and prevalence. The most important result is that the mere dynamics of the HIV epidemic is sufficient to produce rather complex trends in the incidence and prevalence statistics. In contrast, we highlight some previous work (based on traditional mathematical models) that attempt to explain distinctive trends in the reported data, e.g. by the introduction of particularly effective treatments, that we consider unsupported. As a corollary we also substantiate the much-debated paradox that the availability of HAART likely has a negative impact on the evolution of HIV and AIDS. We argue that simulations used for explanation or prediction of observed trends should incorporate more realistic models for both the population and the epidemic than is currently done.

Chapter VII

FUTURE WORK

We plan to investigate various graph properties in terms of the underlying statistical structure of the Kronecker algorithm presented in this dissertation - in particular community structure and diameter. These characteristics are vital for realistic simulations of epidemics, and we expect that the knowledge from this research can also be used to generalize the Kronecker algorithm.

Furthermore, we would like to explore different generalization strategies of the Kronecker algorithm. In the current algorithm the number of categories and their probability distributions are fixed; this leads e.g. to a very irregular degree distribution for larger graphs, and a self-similar expected community-structure. If a different number of categories and degree pmfs can maintain all relevant graph properties, produce a more regular degree distribution and support arbitrary hierarchical community-structure, then population networks can be modeled mor realistically. For instance, the world-wide population is divided into continents (with much stronger connectivity within the continent), continents are divided into countries, etc.

Dynamic assortativity, i.e. choosing which node to connect to is also based on its infection status and type, can be important for the propagation of epidemics as well. For instance, starting a sexual relationship with an infected patient is usually less likely compared to a healthy person. We expect to add dynamic assortativity as follows. Firstly, the probability of choosing a category is 'corrected' in each time step based on its node variable statistics. Within a category, a node's type and status are then chosen and a node label having those values is drawn. A disadvantage of this method is the need for additional $\mathcal{O}(N)$ of storage.

As in many research projects, we have focused on performing simulations of only one or a few different models. However, most parameters are typically estimates at best and of some parameters may be no data available at all. We expect that our framework can be expressive and computationally efficient enough to perform parameter-searching techniques in the space of all model parameters, with a 'fitness' based on emergent statistics such as incidence, prevalence, death rate, varying average degree etc. Assuming that most relevant influences are accounted for in the models, the result's parameters can provide important medical knowledge and the model itself is likely to have predictive power.

Lastly, we are already exploring other possible areas of application of the simulator, such as explaining the evolution of heterogeneous animal populations and the propagation of information in terrorist networks. More generally, we expect that SEECN's paradigm of modeling a population network and propagating a 'virus' can be applied to many different areas.

Appendix A

A LISTING OF THE SIMULATION MODEL PARAMETERS

The parameters that we list here are used in the second simulation of chapter 5, in section 5.5. To obtain the parameters for the first simulation (section 5.4), the progression probability of an untreated HIV patient to being treated becomes zero and the annual expected interaction count (37.5) becomes 30. For brevity we do not specify a node's type in this appendix, because all types are "male" exclusively. A time step in the simulation corresponds to 3 months. Most model parameters are based on [1] and [29].

The separation of distinct infection stages is based on that of [29], and are the following:

Healthy	The node has no virus and does not infect others.
HIV-A	Acute HIV; the first three months after infection.
HIV-B-U	Incubation stage, without treatment.
HIV-B-T	Incubation stage, with treatment.
AIDS	Acquired immunodeficiency syndrome; does not infect others.

The purpose of this appendix is to enable others to use the exact same model parameters as in our simulations.

A.1 Prior probability of status

The prior probability of a node having a given status is used when a new node is created and added to the graph. We assumed that 5% of the homosexual community

Table 1: The prior probability of a new node being assigned a given status.

Status	**Prob.**
Healthy	0.95
HIV-A	0.00125
HIV-B-U	0.04875
HIV-B-T	0.0
AIDS	0.0

Table 2: The directed infection probabilities given the status of both nodes, and the new status of the affected node.

Source status	**Original status**	**New status**	**Prob.**
HIV-A	Healthy	HIV-A	$1-(1-0.05\cdot 0.132)^{37.5/4}$
HIV-A	Healthy	Healthy	$(1-0.05\cdot 0.132)^{37.5/4}$
HIV-B-U	Healthy	HIV-A	$1-(1-0.05\cdot 0.0066)^{31.875/4}$
HIV-B-U	Healthy	Healthy	$(1-0.05\cdot 0.0066)^{31.875/4}$
HIV-B-T	Healthy	HIV-A	$1-(1-0.05\cdot 0.0031)^{37.5/4}$
HIV-B-T	Healthy	Healthy	$(1-0.05\cdot 0.0031)^{37.5/4}$

was infected with HIV, of which a fraction of 1/40 was in the acute state. See table 1.

A.2 Infection probabilities

The probability that a particular node with given 'source status' infects another node with given 'original status', which then changes to 'new status', is non-zero in the following cases and zero otherwise, unless the original status and new status are equal: those cases are set to 1.

The decrease in number of risky sexual contacts for untreated infected nodes is due to the HIV diagnosis in 60% of them. Treated nodes are assumed to be able to resume sexual activity. Successful treatment reduces the infectivity by about 75%, and 30% of the treatments fail.

Table 3: The probability of progression from an original status to a new status per time step, after the minimum progression time.

Original status	**New status**	**Prob.**
HIV-A	HIV-B-U	1.0
HIV-A	HIV-A	0.0
HIV-B-U	HIV-B-T	0.0153
HIV-B-U	AIDS	$0.9847 \cdot 0.9$
HIV-B-U	HIV-B-U	$0.9847 \cdot 0.1$
HIV-B-T	AIDS	0.9
HIV-B-T	HIV-B-T	0.1

A.3 Progression probabilities and minimal durations

Each node is considered for progression in each time step, and progresses to a given new status with probability listed in table 3, unless its current status has not yet lasted for the minimum progression time. During this time, a node is not progressed to any other status. The only exception is the 'progression' of an untreated HIV-patient to being treated, which can occur at any time in the incubation period. We assume that HIV is never diagnosed in the first three months after infection.

All cases that are not listed in table 3 are one if the two statuses are equal, and zero otherwise. The treatment uptake is calculated based on an expected treatment time of 70% of a node's total time of HIV infection [1]. Further, a probability less than 1 for progressing to AIDS incurs a lower median than the average progression time, consistent with [18].

A node's minimum progression time is drawn from a normal distribution with parameters depending on the node's status, given in table 4. Because of the difficulty in estimating the incubation period for AIDS, the reported averages differ greatly. We have assumed a mean of 13 years for untreated patients, and 25 years for a patient that is treated immediately after the acute HIV phase; correcting for the expected time it takes for a node to become treated, the expected incubation period for a treated node becomes 22 years. The standard deviations are estimates, or exceedingly small

Table 4: The parameters for the normal distributions from which the minimum progression times are drawn, indexed on a node's status.

New status	**Mean time**	**Std. dev.**
Healthy	0	$1/\infty$
HIV-A	0	$1/\infty$
HIV-B-U	$4 \cdot 13$	$4 \cdot 2$
HIV-B-T	$4 \cdot 22$	$4 \cdot 3$
AIDS	0	$1/\infty$

Table 5: Expected time before a node with a given status is replaced in the network.

Status	**Prob.**
Healthy	40 years
HIV-A	40 years
HIV-B-U	40 years
HIV-B-T	40 years
AIDS	2 years

if no minimum progression time should be used.[1] These progression times are based on the model presented in [29].

A.4 Node replacement probabilities

In each time step a node is considered for replacement, i.e. removal followed by an immediate addition of a new node. The probabilities are chosen based on the expected time that a node is sexually active and are constant over the simulation. In other words, the time before a node's removal follows a geometric distribution. We find the expected time before removal a more intuitive measure and list them in table 5.

A.5 Edge addition and deletion probabilities

In our simulations we assume no impact on connectivity for a node based on its status. (For AIDS patients, the transmission probabilities are simply set to zero.) Combined with the fact that SEECN does not support dynamic assortativity, the

[1] Note that a minimum progression time of zero or one has no effect, since the minimum time for a node to be considered again for progression is one time step.

probability of an edge being added to the graph is the same for all nodes and equals $1 - pow(1 - 0.5, 1/4)$. The number of times a consideration for adding an edge takes place is drawn from the node's degree pmf. Lastly, inherent in the simulator's implementation, each edge to be added is discarded with probability 0.5 because the nodes at both sides of a possible edge consider to add it and do so with equal probability.

The probability that an existing edge is deleted is also invariant in a node's status, and equals $1 - \sqrt{1 - 0.5} = 0.293$ because each existing edge is considered for removal by both sides. Obviously a node's existing degree can also be viewed as being drawn from its degree pmf, so these two processes balance each other over time and maintain the expected degree.

Appendix B

CALCULATION OF THE KRONECKER PARAMETERS FOR A HOMOSEXUAL COMMUNITY

Firstly, we suppose a power-law degree distribution for the male homosexual community with exponent $\gamma = 1.6$ [23]. Further, a typical reported average number of homosexual relationships per year is around 7 [26], although the number of casual sexual encounters is much higher. We assume that most casual encounters are safe and therefore have limited impact on the propagation of HIV [28]. This yields a maximum degree of $\max_k = 120$. Lastly we take our population size to be 2^{18}, which appears representative for homosexual communities in small countries or large cities.[1]

We find $\tilde{a} + \tilde{b} \approx 0.7$ of the normalized RMAT parameters $\{\tilde{a}, \tilde{b}; \tilde{c}, \tilde{d}\}$ using the binomial cascade presented in [10]. Further conjecturing a common $2 : 3$ relative community strength (see e.g. [11]), using $b = c$ for undirected graphs and combining the expected total edge count of $7 \cdot 2^{18}/2$ with eq. (4), the Kronecker initiator matrix becomes

$$\Theta = \begin{pmatrix} 0.954 & 0.632 \\ 0.632 & 0.051 \end{pmatrix}.$$

These parameters seem to be a good estimate of a homosexual community. In particular, [25] constructed a network representation of a homosexual community based on reported data and observed a big component and unconnectedness using visualization software; according to [19], our Kronecker parameters create indeed a big component and unconnectedness with high probability[2].

[1] An example of which is San Francisco, the area of focus of [24] and [14] against which we compare our results and conclusions.

[2] The authors define 'high probability' as $1 - o(1)$.

REFERENCES

[1] Aalen, O. O., Farewell, V. T., de Angelis, D., Day, N. E., and Gill, O. N., "A Markov model for HIV disease progression including the effect of HIV diagnosis and treatment: Application to AIDS prediction in england and wales," *Statistics in medicine*, vol. 16, no. 19, pp. 2191–2210, 1997.

[2] Aalen, O., Farewell, V., De Angelis, D., Day, N., and Gill, O., "New therapy explains the fall in AIDS incidence with a substantial rise in number of persons on treatment expected," *AIDS*, vol. 13, no. 1, pp. 103–108, 1999.

[3] Ahmed, E. and Agiza, H. N., "On modeling epidemics including latency, incubation and variable susceptibility," *Physica A: Statistical and Theoretical Physics*, vol. 253, no. 1-4, pp. 347–352, 2003.

[4] Albert, R. and Barabási, A.-L., "Topology of evolving networks: Local events and universality," *Phys. Rev. Lett.*, vol. 85, pp. 5234–5237, Dec 2000.

[5] Albert, R., Jeong, H., and Barabasi, A.-L., "The diameter of the world wide web," *Nature*, vol. 401, p. 130, 1999.

[6] Baggaley, R., Ferguson, N., and Garnett, G., "The epidemiological impact of antiretroviral use predicted by mathematical models: a review," *Emerging themes in epidemiology*, vol. 2, 2005.

[7] Barabási, A. L. and Albert, R., "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–519, 1999.

[8] Brookmeyer, R., "Reconstruction and future trends of the AIDS epidemic in the United States," *Science*, vol. 253, no. 5015, pp. 37–42, 1991.

[9] Capasso, V., *Mathematical structures of epidemic systems*. Lecture Notes in Biomathematics No. 97, Springer-Verlag, 1993.

[10] Chakrabarti, D., Zhan, Y., and Faloutsos, C., "R-MAT: A recursive model for graph mining," in *SIAM International Conference on Data Mining*, 2004.

[11] Chakrabarti, D. and Faloutsos, C., "Graph mining: Laws, generators, and algorithms," *ACM Comput. Surv.*, vol. 38, no. 1, p. 2, 2006.

[12] Dye, C. and Gay, N., "Epidemiology: Modeling the SARS epidemic," *Science*, vol. 300, no. 5627, pp. 1884–1885, 2003.

[13] Eubank, S., Guclu, H., and Kumar, V., "Modelling disease outbreaks in realistic urban social networks," *Nature*, vol. 429, no. 6988, pp. 180–184, 2004.

[14] Katz, M. H., Schwarcz, S. K., Kellogg, T. A., Klausner, J. D., Dilley, J. W., Gibson, S., and McFarland, W., "Impact of highly active antiretroviral treatment on HIV seroincidence among men who have sex with men: San Francisco," *Am J Public Health*, vol. 92, no. 3, pp. 388–394, 2002.

[15] Leskovec, J. and Faloutsos, C., "Scalable modeling of real graphs using Kronecker multiplication," in *ICML '07: Proceedings of the 24th international conference on Machine learning*, (New York, NY, USA), pp. 497–504, ACM, 2007.

[16] Leskovec, J., Kleinberg, J., and Faloutsos, C., "Graph evolution: Densification and shrinking diameters," *ACM Trans. Knowl. Discov. Data*, vol. 1, no. 1, p. 2, 2007.

[17] Leskovec, J., Chakrabarti, D., and Faloutsos, C., *Realistic, Mathematically Tractable Graph Generation and Evolution, using Kronecker multiplication*, pp. 133–145. Springer Berlin, 2005.

[18] Longini, I., Clark, W., Byers, R., Ward, J., Darrow, W., Lemp, G., and Hethcote, H., "Statistical analysis of the stages of HIV infection using a Markov model," *Statistics in medicine*, vol. 8, pp. 831–43, July 1989.

[19] Mahdian, M. and Xu, Y., *Stochastic Kronecker Graphs*. Lecture Notes in Computer Science, Springer Berlin, 2007.

[20] Maurer, J., "Boost C++ Libraries - Boost Random Number Library," July 2008. `http://www.boost.org/doc/libs/1_35_0/libs/random/index.html`.

[21] Meyers, L. A., Pourbohloul, B., Newman, M. E. J., Skowronski, D. M., and Brunham, R. C., "Network theory and SARS: Predicting outbreak diversity," *Journal of Theoretical Biology*, vol. 232, pp. 71–81, 2005. http://www-personal.umich.edu/ mejn/papers/mpnsb.pdf.

[22] Moore, C. and Newman, M. E. J., "Epidemics and percolation in small-world networks," *Phys. Rev. E*, vol. 61, pp. 5678–5682, May 2000.

[23] Schneeberger, A., Mercer, C., Gregson, S., Ferguson, N., and Nyamukapa, C., "Scale-free networks and sexually transmitted diseases: a description of observed patterns of sexual contacts in Britain and Zimbabwe," *Sexually Transmitted Diseases*, vol. 31, no. 6, pp. 380–387, 2004.

[24] Schwarcz, S., Kellogg, T., McFarland, W., Louie, B., Kohn, R., Busch, M., Katz, M., Bolan, G., Klausner, J., and Weinstock, H., "Differences in the temporal trends of HIV seroincidence and seroprevalence among sexually transmitted disease clinic patients, 1989-1998: Application of the

serologic testing algorithm for recent HIV seroconversion," *Am. J. Epidemiol.*, vol. 153, no. 10, pp. 925–934, 2001.

[25] Sloot, P. and Ivanov, S., "Stochastic simulation of HIV population dynamics through complex network modeling," *International Journal of Computer Mathematics*, 2007.

[26] Smith, T. W., "Adult sexual behavior in 1989: Number of partners, frequency of intercourse and risk of AIDS," *Family Planning Perspectives*, vol. 23, no. 3, pp. 102–107, 1991.

[27] Wang, Y., Chakrabarti, D., Wang, C., and Faloutsos, C., "Epidemic spreading in real networks: An eigenvalue viewpoint," 2003.

[28] Xiridou, M., Geskus, R., de Wit, J., Coutinho, R., and Kretzschmar, M., "The contribution of steady and casual partnerships to the incidence of HIV infection among homosexual men in Amsterdam," *AIDS*, vol. 17, no. 7, 2003.

[29] Xiridou, M., Geskus, R., de Wit, J., Coutinho, R., and Kretzschmar, M., "Primary HIV infection as source of HIV transmission within steady and casual partnerships among homosexual men," *AIDS*, vol. 18, no. 9, 2004.