Scalable, Efficient Epidemiological Simulation*

Stephen Eubank
D-2, MS-M997
Los Alamos National Laboratory
Los Alamos, NM 87545 USA
eubank@lanl.gov

ABSTRACT

We describe the design and implementation of a system for simulating the spread of disease among individuals in a large urban population over the course of several weeks. In contrast to traditional approaches, we do not assume uniform mixing among large sub-populations or split the population into spatial or demographic subpopulations determined *a priori*. Instead, we rely on empirical estimates of the social network, or contact patterns, that are produced by TRANSIMS, a large-scale simulation of transportation systems.

Categories and Subject Descriptors

I.6.5 [**Computing Methodologies**]: Model Development; J.3 [**Computer Applications**]: Biology and Genetics

General Terms

Design, Algorithms

Keywords

Epidemiology, individual based, parallel discrete event simulation

1. INTRODUCTION

The course of an epidemic is determined largely by the patterns of contacts among individuals. These contact patterns can be represented as a time dependent weighted graph whose vertices represent people. Two vertices are connected by an edge if the corresponding people are in contact. The edges can be weighted to represent factors important in the transmission of disease, such as the duration of contact. In this representation, spread of a disease becomes a problem of diffusion on a time-dependent graph.

Because the graphs in question are so large (millions of vertices and edges), any technique for analyzing an epidemic at the individual level will be very expensive. Compounding the difficulty is the time-dependent nature of the graphs, which change on a time scale of seconds or minutes, yet need to be analyzed over a several week period. Furthermore, the nature of the problem makes it very difficult or unethical to conduct experiments on real populations, especially at the large scales we need to understand.

Hence we turn to simulation. Distributed simulation of local interactions results in knowledge about exactly those global properties of the graph relevant to the spread of a specific disease. We can use the simulation as an experimental test-bed for analyzing the effects of interventions (for example, vaccination or quarantine) targeted at specific geographic or demographic sub-populations.

The epidemiological simulation itself does not necessarily provide a better understanding of the complex interactions between disease and social network. It will not, for instance, suggest what intervention strategies to try. It is more likely that such understanding will come from a graph theoretical analysis of the social network.[2] Yet even *evaluating* many of the graph properties likely to be important for disease transmission is NP-hard.[5] Analyzing the social network's structure in enough detail to understand how it interacts with disease is likely to be even harder. Hence one goal of the simulation design is to develop distributed, local algorithms for approximating and analyzing global properties of large graphs, similar in spirit to those Kleinberg proposes for analyzing the World Wide Web.[12, 11]

2. REQUIREMENTS

This simulation falls into the class of Sequential Dynamical Systems (SDS).[14] An SDS has three components:

1. A dependency graph. In our case this graph is a set of cliques, one for each location, in which any two people who are present can interact.

2. A local mapping. The local mapping here is determined by details of the disease transmission and progression.

3. An update order. Here we choose parallel update among all people present at a location. Fortunately, for most diseases the time scale for transmission is short compared to that for progression within a host. Thus the dynamics of transmission and progression can be separated, resulting in fairly simple models for each.

The SDS that represents epidemiology is particularly rich:

*LAUR-01-5513

SAC 2002 Madrid, Spain

its dependency graph is time-dependent and its local mapping is stochastic.

The principal requirement for the simulation is scalability – we intend to simulate urban regions with millions of people over time scales of weeks. If possible, we would like to be able to make policy decisions in real time in the face of an outbreak. This requires sensitivity studies exploring the effects of different policies with multiple (tens or hundreds) of simulation runs. Thus, the simulation and all overhead associated with initializing different scenarios should run several orders of magnitude faster than real time. Our target computing environment is a cluster of dozens to thousands of computing nodes.

Another important requirement is the ability to identify specific geographic and demographic pathways along which disease spreads. For example, we might find that downtown fast-food establishments or 6-year-old children are on the critical transmission path. This imposes a very disaggregated level of description on the simulation. In our case, we take this to the extreme level of individual people and their activity locations.

The system must be modular with respect to diseases, contact patterns, and transmission probabilities. That is, it must be able to handle several different diseases with different dynamics (bacterial versus viral, for example) easily. Similarly, it should be able seamlessly to integrate several possible sources of the underlying social networks.

3. DESIGN

The design we have chosen is a distributed discrete event simulation. Data flow in the simulation is sketched in the block diagram of Figure 2 and described in more detail below. Each individual is represented by an object that contains a subset of the available demographic information as well as his or her state of health. Each computational node is responsible for all the interactions at a subset of (geographical) locations. Individuals are passed among computational nodes via messages as they go about their daily activities.

3.1 Estimating Contact Patterns

Though not strictly speaking part of the epidemiology simulation, it is important to understand where the social network we use comes from. One of the distinguishing features of the current work is the level of detail it makes available. All this detail relies on corresponding detail in the input data. Our approach in this and other simulation projects is to disaggregate to the individual level; allow individuals to interact using as simple as possible a model that produces results consistent with observed results; and re-aggregate as necessary for comparison with observations. The operations of aggregating and disaggregating do not commute with applying local interactions. That is, the equivalent interaction in an aggregated picture, if it exists, would be very different from that in the disaggregated picture. The whole program of "renormalization" in physics is an attempt to understand how a local interaction changes into an effective long-range interaction as the system is aggregated.[8] In cases where there is no global symmetry in the problem, it can be very difficult to determine the equivalent interaction. In these cases, disaggregate modeling is the only feasible solution technique.

Epidemiology is only one of several large-scale simulations currently under development that rely on detailed estimates of the social network in a large urban area. Until recently, it has been difficult to obtain these estimates. Typically, certain classes of random graphs (in recent times, especially small-world networks) have been postulated as good representatives. Alternatively, observations of networks in small samples have been extrapolated to whole populations. As the possible benefits to society of these various simulations are demonstrated, they will drive development of better estimates of social networks.

Already, the Transportation Analysis and Simulation System (TRANSIMS) developed at Los Alamos National Laboratory produces estimates of the social network in a large urban area based on the assumption that the transportation infrastructure constrains people's choices about what activities to perform and where to perform them.[1] See Figure 1. A synthetic population is endowed with demographics matching the joint distributions given in census data. Observations are made on the daily activity patterns of several thousand households (survey data). These patterns are used as templates and associated with synthetic households with similar demographics. The locations at which activities are carried out are estimated taking into account observed land use patterns, travel times, and dollar costs of transportation. The estimated locations are fed into a routing algorithm to find minimum cost paths through the transportation infrastructure consistent with constraints on mode choice. An example constraint might be: "walk to a transit stop, take transit to work using no more than 2 transfers and no more than 1 bus". After routing several tens of millions of trips in this way on a transportation network with 120,000 links, we simulate the traffic induced when everyone tries to execute their plans simultaneously. The simulation resolves traffic down to 7.5 meters and times down to 1 second. It provides an updated estimate of link costs, including the effects of congestion, to the Router and location estimation algorithms, which produce new plans. This feedback process continues iteratively until convergence to a steady state in which no one can find a better path in the context of everyone else's decisions. The resulting traffic patterns are matched to observed traffic.

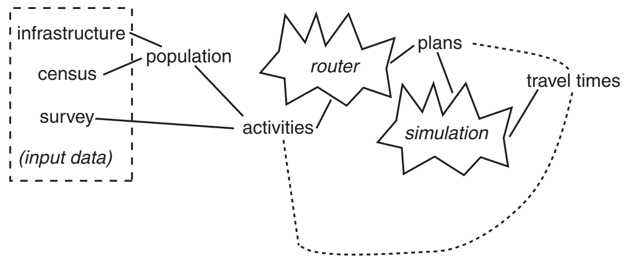

Figure 1: Data flow in the TRANSIMS simulation system, proceeding from left to right. Input data comes from the U.S. census and metropolitan planning organizations. We generate a synthetic population whose demographics match the census; give each household an appropriate set of activities; plan routes through the network; and estimate the resulting travel times. The dotted lines represent feedback pathways, along which data flows from right to left, in the system.

The TRANSIMS system is also an SDS at many different levels. The traffic simulation is an obvious one, but the relation between the router and the traffic simulation is also an SDS. The system consists of about 200,000 lines of mostly C++ code and runs on a variety of UNIX platforms. Production runs for a case study in Portland, Oregon are being made on a Linux cluster using 48 CPUs for the traffic simulator and 64 for the router. The router takes roughly 20 hours to generate trips for the entire population; the traffic microsimulator runs about 3 times faster than real-time, on average. At the current time, no information is available regarding scalability. The current number of processors was chosen to keep each process's memory requirement comfortably below 1 Gigabyte.

3.2 Data Flow in the Epidemiological Simulation

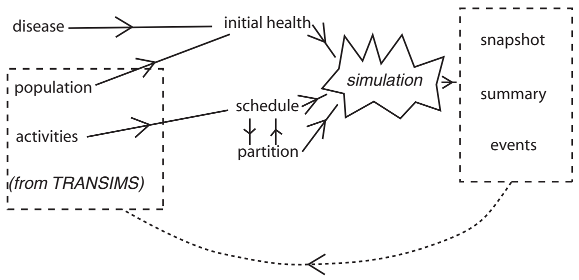

Figure 2: Data flow in the epidemiology simulation system. Input data comes from two sources: the user's disease model and information about the social network. Stand-alone tools operate on the disease model and the population's demographics to produce the initial state of health for everyone in the simulation. Another tool converts a list of activities and locations organized by person into a schedule of events (primarily arrivals and departures) organized by location. The final preparation step estimates an optimal partition of resources among computational nodes. The simulation itself executes events in strict time order and propagates disease in accordance with the user's disease model. It produces three kinds of output: snapshots at specified intervals for animation, statistical summaries of the simulation, and sets of disease-related events. In the near future, we will add the feedback pathway shown by the dotted line, allowing a person's state of health to affect his or her activities.

Every individual's initial state of health is specified before the simulation begins. Tools for easily implementing common initializations depending on demographics are provided. For instance, vaccinating a fraction of those under 5 years old is accomplished by manipulating the parameters describing their immune response. Users can, of course, write their own tools.

Work is distributed among the computational nodes by partitioning the set of locations statically. We create a graph with vertices weighted by the number of visitors to a location and edges weighted by the number of people traveling between locations. Any standard partitioning algorithm can be used to assign locations a partition. We currently use the METIS library[9].

The simulation simply moves people from location to location according to their predetermined schedules, keeping track of their state of health as they move, and allows them to interact.

3.3 Component Models

The form of the disease model does not affect the mechanics of the simulation itself. Our goal, as always, is to use the simplest model possible that can be shown to give results consistent with observation. We are exploring two different representations for a person's state of health. The most useful of these seem to be a "compartment" model (essentially a finite state machine) and a "load" model.

Compartment models track the evolution of a few possible states, such as "Susceptible", "Infected", or "Recovered". Their names are typically acronyms derived from the allowed states - SIR in the instance above. Our compartment models are derived from traditional epidemiological models, except that we assign states to individuals instead of entire subpopulations, and we interpret transition rates as probabilities for an individual in one state to move to another instead of the fraction of a subpopulation that moves to another state.

Load models provide a more detailed description of particular diseases. They represent an individual's disease state as a floating point "load" that could be taken as, for example, the viral concentration that would be detected in a throat swab. The load grows or decays exponentially with time. The model must specify several independent parameters for each person:

- The rate at which the load will grow or decay in a host. This will, in general, depend upon time since infection, vaccination status, and overall state of immune system.

- The rate at which the load is shed into or absorbed from the environment. This might depend on demographic factors such as age or cultural factors such as closeness of contact or hand-washing frequency.

- The effects of a given load. Some people, for example, may exhibit symptoms at lower thresholds than others. The thresholds might be used to project a load into a small set of states as in the compartment models.

One advantage of the load model is that it very naturally captures the possibility of non-human or even non-living hosts. A conference room that has been recently visited by many people with colds can hold a slowly-decaying load; or an establishment that serves food can be associated with a higher than average rat population, that in turn may carry flea-borne disease. Of course, not all these parameters (and certainly not their distributions over demographics) are available for all diseases of interest. Nonetheless, the model is capable of simulating the disease process to whatever degree of fidelity is warranted by available data.

The design of the simulation allows multiple disease states to be associated with each person.[13] The number used is determined at run time. Currently, there is no interaction among disease states, but this may be explored. It would provide a means of modeling, for example, distinct strains of influenza circulating during an outbreak. Exposure to one strain may increase or decrease susceptibility to other strains. Even without interaction among different diseases,

person	location	arrival time	departure time
1	A	8:00	17:00
2	A	8:00	12:00
2	B	12:00	16:00
3	C	8:00	14:00
3	B	14:00	17:00

Table 1: A hypothetical set of activities for 3 people visiting 3 locations.

this feature can simulate multiple independent epidemics simultaneously. Thus a single pass through the event schedule, which is the time-consuming aspect of the simulation, can generate multiple independent epidemics for statistical analyses.

3.4 Event Scheduler

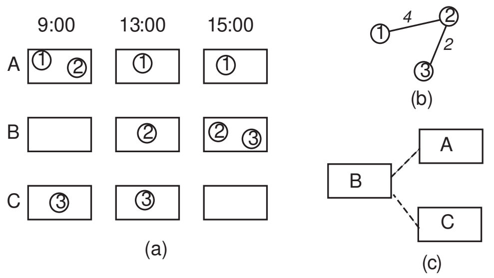

Figure 3: (a) The set of activities from Table 1 is represented here as a time-dependent bi-partite graph. Each of the locations A, B, and C shown as rectangles contains a subset of the people 1, 2, and 3. These time-dependent graphs induce many possible "collapsed" graphs. Here we illustrate two: (b) a weighted, undirected graph in which vertices (people) are connected with an edge if and only if they were in contact at some point during the day. Edges are weighted by the total duration of the contacts; (c) an unweighted, undirected graph in which vertices (locations) are connected if the sets of people at those locations at any time during the day have a non-empty intersection.

The sets of locations visited by every person during the course of a day induces a time dependent social network, or graph of contact patterns. At any instant the graph is composed of a set of complete sub-graphs, one for each location where people are present. For example, Figure 3(a) shows the graphs induced by the set of activities in Table 1. Figure 4 displays the distribution of maximum sizes of the subgraphs corresponding to locations in Figure 3(a), but based on a full activity set. We note that it exhibits power law scaling with an exponent of approximately -2.6.

The novel feature of the epidemiology simulation is its scalability: TRANSIMS provides a set of, on average, 5.5 daily activities for each of 1.6 million people distributed across 140,000 locations. Until very recently, research in TRANSIMS has focused on generating good estimates of the activity patterns for one city. We are just beginning to analyze the resulting social network. Because of the large size and time-dependent nature of the social network, we have concentrated on related time-independent networks, as illustrated in Figure 3(b) and (c). The distribution of vertex degrees for a graph corresponding to Figure 3(c) on the TRANSIMS activity set is shown in Figure 5.

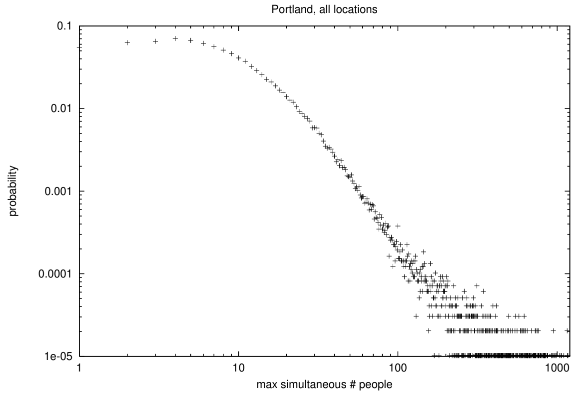

Figure 4: The distribution of maximum number of people at any location during a day.

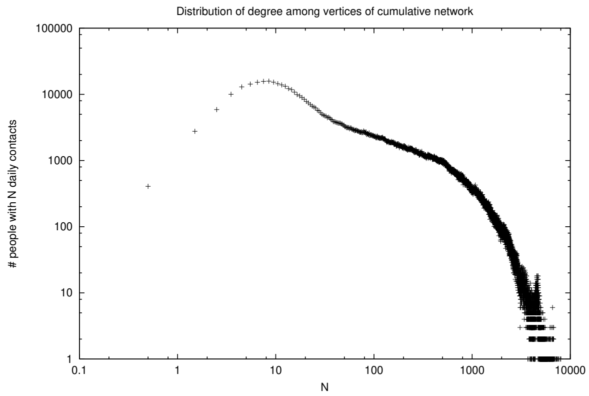

Figure 5: The distribution of vertex degrees in the contact graph collapsed over a day. In other words, the total number of people a person shares any location with over the course of a day.

The epidemiological simulation turns an implicit description of a social network contained in a set of activities into an explicit, time-dependent instantiation of the graph. The simulation maintains a state for every location, consisting of a list of objects representing the people present and a disease load for the location. Events such as "contaminate a location" or "move a person from location x to y" update the state of all affected locations and, if necessary, delete and create edges in the graph.

The current implementation's locations map one-to-one to the locations specified in the input social network. Depending on the resolution of the model that creates the social network, it may be desirable to disaggregate these locations. For example, TRANSIMS places roughly four locations on every city block. Some of these represent a collection of office buildings holding hundreds or thousands of workers. Our simulation in effect treats the entire population of a location as if it were well-mixed. An obvious enhancement to this is to allow hierarchical mixing, with membership in the smallest groups determined stochastically.

The events in the system are related to:

- the underlying graph, for example "individual i moves to location x at time t"
- the disease process, for example "location y becomes contaminated at time t"
- possible interventions, for example "individuals with demographics D are given an anti-viral drug at time t"

There is no dynamic scheduling of events – the time at which each event occurs can be determined before running the simulation. We implement the simulation as a controller that steps through the simulation executing events. Every event at a location or every request for output from a location, can trigger computation to update the current state of all individuals present at the location. There is no optimistic evaluation of future states and thus no roll-back mechanism.

Our prototype has been developed to test the scalability of the design. We have so far used it primarily to calculate certain global properties of the social networks in a distributed, local fashion as described below. We typically "infect" a large fraction of the population in most experiments. In our prototype we have chosen to track each person, regardless of health state, throughout the entire day. Obviously, if only a few people are infected, it is more efficient to track only those people and their contacts. This approach, however, requires global synchronization to determine whether an infected person's contacts are themselves infected or can be initialized at this location. One open question is how much more efficient this would be – it is directly related to the question of how rapidly disease will spread.

4. COMPARISON TO OTHER MODELS

One very popular model in epidemiology has been the compartmental model referred to above.[7, 15] The population is described by the fraction f_i that falls into any of m states. Specifying transition rates $\mathcal{R}_{ij}$ among the states gives a set of rate equations – coupled ordinary differential equations that determine the evolution of the system:

$$\frac{d\vec{f}}{dt} = \mathcal{R}\vec{f}.$$

This representation is well-suited to analytical techniques, determining constraints on the rates that lead to different asymptotic states.

There are, as usual, consequences associated with finite-size populations, which imply discrete rather than continuous evolution. These, however, pale beside the dramatic assumption that lies at the heart of these models: uniform mixing of the population. In our language, it is as if the dynamics take place on a complete (fully-connected) time-invariant graph with equal weights on each edge. The limitations of this assumption are well-known. The ecological literature in particular emphasizes the importance of spatial structure in population dynamics.[10, 3]

Several attempts have been made to relax this assumption, in two principal ways: structured population models and spatial models. Structured population models postulate the existence of several disjoint subpopulations, each of which is uniformly mixed and connected to each of the others by a coupling constant. A typical way to classify people into subpopulations is by age: three or four subpopulations corresponding to pre-school, school-age, adults, and the elderly. There are several problems inherent in this approach:

- It still assumes uniform mixing among large groups of people.
- It introduces parameters, the coupling between subpopulations, whose values can only be guessed at or fit.
- The subpopulations are univariate. There is no way a person could be described as both a member of a family of four and a person who takes mass transit to work.

Spatial models break the population into spatial cells and allow for migration between cells. They could be viewed as a particular structured population, with couplings between subpopulations related to the amount of migration. The spatial structures considered, however, are typically static, regular lattices allowing migration between neighboring cells. These do not capture well the irregular, long-range, time dependent nature of urban social networks. For example, the distribution of numbers of edges between locations in our social network is likely to be at least as complicated as a power-law, whereas for a static two-dimensional grid, every location would have exactly four edges.

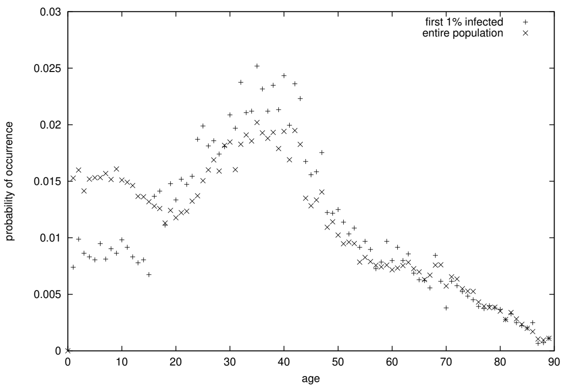

Figure 6: The distribution of ages in the population as a whole and in the first 1% of the population infected in a simulation run.

In contrast, results of an individual-based simulation could be aggregated to *derive* the proper structures for these models. Coupling constants between subpopulations could be estimated directly from the social network. The best variables on which to structure subpopulations should be detectable as natural clusters in simulation results. Figure 6 compares the distribution of ages for the entire population and the first 1% to be infected in a simulation run. Information like this could be used to define appropriate subpopulation structures. However, the problems of allowing non-disjoint structures and of finite-size populations are still best addressed by individual-based simulation.

There have been efforts to create individual-based epidemiological simulations before, but they have generally been limited by the lack of suitable estimates of contact patterns. In particular, Ackerman has performed small-scale simulations using observed contacts among small groups of relatively isolated people – for example, residents of a nursing home.[16] Unfortunately, it is difficult to scale these observational data up to entire urban areas.

Another approach to the problem of estimating contact patterns has been to use instances of particular classes of regular or random graphs. Several classes of random graphs have been proposed based on academic citation networks or World Wide Web hyperlinks. Many of these networks share important properties, such as existence of a giant component[4] or "small-world-ness", in which a certain fraction of edges connect vertices that would otherwise be far apart.[17] But it is difficult to imagine what class of random graph would produce the distribution shown in Figure 5.

5. IMPLEMENTATION

The simulation system has been implemented in C++ using the mpich messaging library[6] and run on both a Sun 14-processor SPARCServer and a Pentium III based Linux cluster of 16 CPUs with ethernet. METIS[9] is used to build the partitions.

The simulation has been run on a social network produced by TRANSIMS that describes one day (a school day) in the life of a city of 1.6 million people with 250,000 activity locations. This single day is repeated many times in the current implementation. Randomization of some activities will be introduced in the future.

To give an appreciation for the scale of this simulation, we display the file sizes in megabytes for this example:

- population - 75
- activities - 1024
- initial health - 25
- events - 500
- collapsed daily graph (see next section) - 12,000

All input files are ASCII text, associated with one or more binary index files. Each index entry contains a file identifier and offset together with one or two key values, such as location id and event timestamp. The entries are sorted by major and minor key and stored in a binary tree. Index files can be merged and sorted without moving the data files to which they point. In addition, index files need not contain an entry for every element of the data file(s). This provides a simple mechanism for distributing events to the relevant processor. Each process holds a different index, containing only the entries relevant for that processor.

Run times are currently approximately 100 times faster than real time with 16 CPUs. A typical run simulating 40 days takes a few hours. Since we are most interested in the course of a single outbreak, not dynamics over many years, this speedup is sufficient for our purposes.

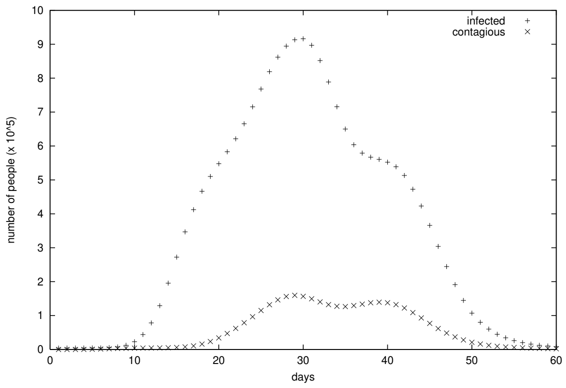

Figure 7: The number of people infected and contagious as a function of time in a simulated outbreak.

Our work to date has focused on development of the simulation and not on analysis of results. However, we have performed a simulation for a highly contagious disease with an incubation period of about 12 days and an contagious period of 10 days beginning about 3 days after the appearance of symptoms. This outbreak is started by contaminating a single location at day 0. The resulting epidemic curves are shown in Figure 7. Note that, in this model, the "contagious" cohort is a subset of the "infected" cohort, rather than disjoint as in a compartmental model.

6. ALGORITHMIC "DISEASES"

So far, this paper has been devoted to simulating epidemiology. But there is another use for this simulation, alluded to briefly above. The spread of disease occurs through local interaction followed by diffusion of information in a local neighborhood – a reaction-diffusion process on a time-dependent graph. This is a model for an important class of computation. Since we already have a flexible scheme for specifying diseases, it is an amusing challenge to see if we can define "diseases" that accomplish useful computation. This must be distinguished from building computer viruses. It is more a question of determining the global semantics of distributed, local mappings.

As an example, consider the problem of collapsing the time-dependent social network into one showing the connectivity for an entire day. Specifically, as in Figure 3c, we construct a graph whose vertices are individuals. There is an edge between two vertices if the corresponding people were ever in the same location simultaneously during the day. Each edge is assigned a weight corresponding to the total duration the two individuals were in contact during the day (summing across locations).

There is a "disease" that calculates the edge weights very naturally. Each person carries an instance of the disease with a unique identifier (think "genetic sequence"). Every person becomes "infected" with the "disease" shed by every other person with whom he or she comes into direct contact. The load is proportional to the amount of time spent in contact and neither grows nor decays when the contact is broken. Each person sheds only his or her own peculiar "disease". At the end of one day, we simply write out the diseases and loads carried by each person to get a description of the edges and weights in the desired graph. This is, in fact, the algorithm used to create Figure 5.

One can easily imagine that there are "diseases" that will calculate shortest path, reachability, diameter, and other useful quantities in a local, distributed fashion. The local algorithms may not be obviously related to the global results. The efficiency of these distributed calculations is related to properties of the interaction graphs. Understanding how actual diseases have evolved to exploit the properties of social networks can give us insight into both the efficiency of evolutionary search strategies and characteristics of the social networks to which they have adapted.

7. REFERENCES

[1] C. L. Barrett, R. J. Beckman, K. P. Berkbigler, K. R. Bisset, B. W. Bush, K. Campbell, S. Eubank, K. M. Henson, J. M. Hurford, D. A. Kubicek, M. V. Marathe, P. R. Romero, J. P. Smith, L. L. Smith, P. L. Speckman, P. E. Stretz, G. L. Thayer, E. V. Eeckhout, and M. D. Williams. TRANSIMS: Transportation Analysis Simulation System. Technical Report LA-UR-00-1725, Los Alamos National Laboratory Unclassified Report, 2001.

[2] B. Bollobas. *Random Graphs*. Academic Press, London–New York, 2001.

[3] R. Durrett and S. A. Levin. The importance of being discrete (and spatial). *Theoretical Population Biology*, 46:363–394, 1994.

[4] P. Erdos and A. Renyi. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17–61, 1960.

[5] M. R. Garey and D. S. Johnson. *Computers and Intractability*. W. H. Freeman, New York, 1979.

[6] W. Gropp and E. Lusk. User's Guide for mpich,a Portable Implementation of MPIVersion 1.2.2. Technical Report, Argonne National Laboratory, http://www-unix.mcs.anl.gov/mpi/mpich/, 2001.

[7] H. W. Hethcote. The Mathematics of Infectious Diseases. *SIAM Review*, 42(4):599–653, 2000.

[8] L. P. Kadanoff. *Statistical Physics: Statics, Dynamics, and Renormalization*. World Scientific, Singapore, 2000.

[9] G. Karypis and V. Kumar. METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices. Technical Report http://www.cs.umn.edu/ karypis, University of Minnesota, Department of Computer Science / Army HPC Research Center, 1998.

[10] M. J. Keeling and C. A. Gilligan. Metapopulation Dynamics of Bubonic Plague. *Nature*, 407:903–906, 2000.

[11] D. Kempe, J. Kleinberg, and A. Kumar. Connectivity and inference problems for temporal networks. *Proc. 32nd ACM Symposium on Theory of Computing*, 2000.

[12] J. Kleinberg. Navigation in a Small World. *Nature*, 406:845, 2000.

[13] W. A. Maniatty, B. K. Szymanski, and T. Caraco. High-performance computing tools for modeling evolution in epidemics. In *Proc. 32nd IEEE Annual Hawaii International Conference on System Sciences, HICSS'99*. IEEE, January 1999. CD ROM Disk, 10 Pages (Abstracts pp. 318-319).

[14] H. S. Mortveit and C. M. Reidys. Discrete, sequential dynamical systems. *Discrete Mathematics*, 226:281–295, 2001.

[15] J. D. Murray. *Mathematical Biology*. Springer-Verlag, Berlin, 1989.

[16] D. Peterson, L. Gatewood, Z. Zhuo, J. J. Yang, S. Seaholm, and E. Ackerman. Simulation of stochastic micropopulation models. *Computers in Biology and Medicine*, 23(3):199–210, 1993.

[17] D. Watts and S. Strogatz. Collective Dynamics of Small-World Networks. *Nature*, 393:440, 1998.